Reference number of working document: **ISO/TC 184/SC1 N 000**

Version description: **V4 - September 2001**

Date: 2001-09-10

Reference number of document: **ISO/WD 14649-12**

Committee identification: ISO/TC 184/SC 1/WG 7

Secretariat: DIN

Industrial automation systems and integration - Physical device control - Data model for Computerized Numerical Controllers -

Part 12:

Process data for turning

Warning

This document is not an ISO International Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an International Standard.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: Working Draft
Document subtype: if applicable
Document stage: Preparation
Document language: English

Information on technical or structural contents are available at the following addresses:

Convener:	Fredrich Glantschnig
	AMT Consulting
Address:	Hohenweg 33a
	CH-5417 Untersiggenthal
Telephone:	+41 (056) 288-2042
	+41 (056) 288-3942
E-mail:	fglantschnig@swissonline.ch

Owners of this part of documents:

	Prof. Suk-Hwan Suh	Stefan Heusinger
	NRL-SNT, POSTECH	ISW, University of Stuttgart
Address:	San 31, Hyoja-Dong, Nam-Gu	Seidenstrasse 36
	Pohang, Kyungbuk, 790-784, KOREA	D-70174 Stuttgart
Telephone:	+82 (54) 279-2196	+49 (0711) 121-2342
FAX:	+82 (54) 279-5998	+49 (0711) 121-2413
E-mail:	shs@postech.ac.kr	stefan.heusinger@isw.uni-stuttgart.de

Contents

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 3.

Draft International Standards adopted by the technical committees are circulated to the member bodies for voting. Publication as an International Standard requires approval by at least 75 % of the member bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this part of ISO 14649 may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights.

International Standard ISO 14649-12 was prepared by Technical Committee ISO/TC 184, *Industrial automation systems and integration*, Subcommittee SC 1, *Physical device control*.

ISO 14649 consists of the following parts, under the general title

Data model for Computerized Numerical Controllers

— Part 1: *Overview and fundamental principles, published as actual DIS Phase 1*

— Part 2: *Language bindings, Fundamentals, will be published as Phase 3*

— Part 3: *Language binding in Java, will be published as Phase 3*

— Part 9: *Glossary, will be published as Phase 3*

— Part 10: *General Process Data, published as actual DIS Phase 1*

— Part 11: *Process Data for Milling, published as actual DIS Phase 1*

— Part 12: *Process Data for Turning, will be published as Phase 3*

— Part 13: *Process Data for EDM, will be published as Phase 3*

— Part 50: *AIM of General Process Data, will be published as Phase 2 as AP238*

— Part 51: *AIM of Process Data for Milling, will be published as Phase 2 as AP238*

— Part 52: *AIM of Process Data for Turning, will be published as Phase 3 as AP238*

— Part 53: *AIM of Process Data for EDM, will be published as Phase 3 as AP238*

— Part 111: *Tools for Milling, published as actual DIS Phase 1*

— Part 121: *Tools for Turning, will be published as Phase 3*

Introduction

Subject of the schema turning, which is described in this paper, is the definition of technology-specific data types representing the machining process for turning operation by lathe or turn-mill center. The information contents of this schema are derived based on the analysis of user requirements for turning operation including complete machining. This structure of the schema is compliant with the currently developed schema ISO 14649 including Part 10 (General process data), and Part 11 (Process data for milling). The machining features defined in this schema are harmonized with ISO 10303 AP224 (Mechanical product definition for process planning using machining features).

Not included in this schema are representation, executable objects, and base classes which are common for all technologies. They are referenced from ISO 10303's generic resources and Part 10 of this standard. The description of process data is done using the EXPRESS language as defined in ISO 10303 Part 11. The encoding of the data is done using ISO 10303 Part 21.

Industrial automation systems and integration — Physical device control — Data model for Computerized Numerical Controllers — Part 12: Process data for turning

1 Scope

This part of ISO 14649 specifies the technology-specific data elements needed as process data for turning. Together with the general process data described in Part 10 of this standard, it describes the interface between a computerized numerical controller and the programming system (i.e. CAM system or shopfloor programming system) for turning. It can be used for turning operations on all types of machines, be it turning machines, lathes or turning machining centers. The scope of this part does not include any other technologies, like milling, grinding, contour cutting, or EDM. These technologies will be described in further parts of this standard.

2 Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this part of ISO 14649. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. However, parties to agreements based on this part of ISO 14649 are encouraged to investigate the possibility of applying the most recent editions of the normative documents indicated below. For undated references, the latest edition of the normative document referred to applies. Members of ISO and IEC maintain registers of currently valid International Standards.

ISO 10303 Part11: Industrial automation systems and integration: Product data and exchange - Description methods: the EXPRESS Language Reference Manual (IS)

ISO 10303 Part21: Industrial automation systems and integration: Product data and exchange - Implementation methods: Clear text encoding of exchange structure (IS)

ISO 10303 AP 203 Application protocol: Configuration controlled designs (IS)

ISO 10303 AP 214 Application protocol: Core data for automotive mechanical design process (IS)

ISO 10303 AP224 Application protocol: Mechanical product definition for process planning using machining features (IS)

ISO 14649 Part 10: Data model for computerized numerical controllers – General process data (DIS)

ISO 14649 Part 11: Data model for computerized numerical controllers - Process data for milling (DIS)

3 Terms and definitions

For the purposes of this part of ISO 14649, the terms and definitions given in Part 10 of this standard and the followings apply.

3.1 Finishing

A machining operation used to cut a part. The finishing operation usually follows a roughing operation. The goal of finishing is to reach the surface quality required, cf. roughing.

3.2 Roughing

A machining operation used to cut a part. While the aim of roughing is to remove large quantities of material in a short time, the surface quality is usually not important. The roughing operation is usually followed by a finishing operation, cf. finishing.

4 Symbols (and abbreviated terms)

No symbols defined in this part.

5 Process data for turning

5.1 Header and references

The following listing gives the header and the list of entities which are referenced within this schema.

```
SCHEMA turning_schema;

(*
   Version : 04
   Date    : 09.10.2001
   Author  : ISO TC184/SC1/WG7
   Contact : Suk-Hwan Suh (shs@postech.ac.kr) or
             Heusinger (stefan.heusinger@isw.uni-stuttgart.de)
*)


(* ********************************************************************** *)
(* Types from machining_schema                 ISO 14649-10               *)
(* ********************************************************************** *)

REFERENCE FROM machining_schema(
    axis2_placement_3d,
    bounded_curve,
    cartesian_point,
    direction,
    label,
    length_measure,
    machine_functions,
    machining_feature,
    machining_operation,
    machining_strategy,
    material,
    poket_bottom_condition,
    positive_ratio_measure,
    plane_angle_measure,
    pressure_measure,
    property_parameter,
    radial_direction,
    rot_direction,
    rot_speed_measure,
    slot_end_type,
    speed_measure,
```

```
technology,
time_measure,
toleranced_length_measure,
toolpath_list,
user_defined_function,
);
```

5.2 Manufacturing features for turning

The base class of all of features used for turning will be the class turning_feature. The turning_feature itself is derived from the two5D_manufacturing_feature described in “Part 10: General Process Data”.

The following entity shows the definition of the two5D_manufacturing_feature. Further details and especially the representation in EXPRESS-G language can be found in Part 10.

```
ENTITY two5D_manufacturing_feature
  ABSTRACT SUPERTYPE OF (ONEOF(machining_feature,
    replicate_feature,compound_feature, turning_feature));
  feature_placement : axis2_placement_3d;
  feature_tolerance : tolerance;
  its_datum : SET [0:?] OF datum;
END ENTITY;
```

Fig. 1: two5D_manufacturing_feature used as base class for the turning_feature

5.2.1 Turning feature

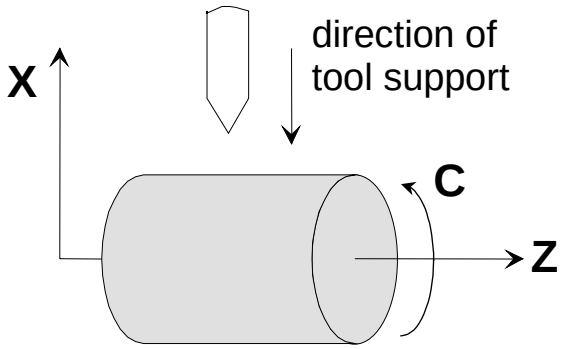


Fig. 2: Turning operation incorporated in turning feature

The entity turning_feature is the abstract base class for all features used for turning. The defined turning features are classified geometric shapes that can be obtained by turning the cylindrical workpiece with 2-axis (x and z) operation or 3-axis (x, z, and c) operation (Fig. 2). Currently, the following features are included: knurl, cut_in, outer_round, revolved_feature, end_face. The features are defined in close resemblance to ISO 10303-224 except for cam and end_face, which is newly defined by reflecting the 3-axis turning operation.

```
ENTITY turning_feature
   ABSTRACT SUPERTYPE OF (ONEOF(knurl, outer_round, revolved_feature, cut_in,
         end_face))
   SUBTYPE OF (two5D_manufacturing_feature);
    material_side : OPTIONAL direction;
```

```
END_ENTITY;
```

material_side: The material_side specifies the material direction. The direction of removal indicates the direction that the material will be removed from the part.

5.2.2 Knurl

A knurl is a type of turning_feature that is a scoring pattern made by a series of small ridges or beads on a metal surface. A specially designed tool is used for making the knurl. Four types of pattern are defined: 1) straight_knurl, 2) diagonal_knurl, 3) diamond_knurl, and 4) user_defined_knurl. The user_defined_knurl includes catalogue_knurl defined in ISO 10303 AP224 and other patterns.

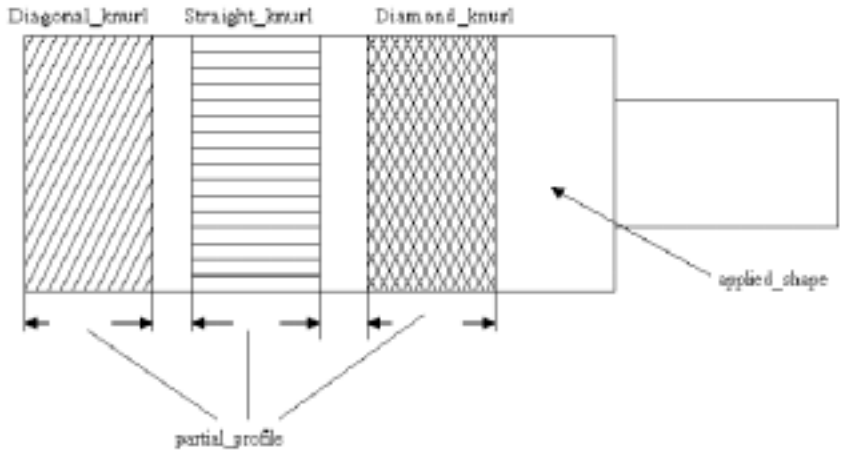


Fig. 3: Knurl

```
ENTITY knurl
  ABSTRACT SUPERTYPE OF (ONEOF(straigt_knurl, diagonal_knurl, diamond_knurl,
    catalogue_knurl))
  SUBTYPE OF (turning_feature);
  base_feature :          turning_feature;
  tooth_depth :           OPTIONAL toleranced_length_measure;
  diameter_pitch :        OPTIONAL toleranced_length_measure;
  root_fillet :           OPTIONAL plane_angle_measure;
  number_of_teeth :       OPTIONAL INTEGER;
  major_diameter:         OPTIONAL toleranced_length_measure;
  nominal_diameter:       OPTIONAL toleranced_length_measure;
END_ENTITY;
```

base_feature: this attribute is a reference to a feature, on which the knurl is applied to.

tooth_depth: the tooth_depth specifies the depth from the crest of a tooth to the point where two teeth intersect.

diameter_pitch: the diameter_pitch specifies the ratio of the number of teeth in the circumference to the nominal diameter.

root_fillet: the major_diameter specifies the dimension of a radius between teeth on a knurling tool.

number_of_teeth: the major_diameter specify the size of the part before a knurl is applied to it.

major_diameter: the major_diameter specify the size of the part before a knurl is applied to it.

minor_diameter: the minor_diameter specify the size of the part after a knurl has been applied.

5.2.2.1 Straight knurl

A straight_knurl is a type of knurl that is the knurl scoring that is parallel to the axis of the scored surface (z-axis).

```
ENTITY stratight_knurl
  SUBTYPE OF (knurl);
END_ENTITY;
```

5.2.2.2 Diagonal knurl

A diagonal_knurl is a type of knurl with helical cuts at an angle about the axis of a surface.

```
ENTITY diagonal_knurl
  SUBTYPE OF (knurl);
  helix_angle:    plane_angle_measure;
END_ENTITY;
```

helix_angle: the helix_angle specifies the angle the knurl pattern makes with the orientation axis of an applied to surface.

5.2.2.3 Diamond knurl

A diamond_knurl specifies the angle the knurl pattern makes with the orientation axis of an applied to surface.

```
ENTITY diamond_knurl
  SUBTYPE OF (knurl);
  helix_angle:    plane_angle_measure;
END_ENTITY;
```

helix_angle: the helix_angle specifies the angle the knurl pattern makes with the orientation axis of an applied to surface.

5.2.2.4 Catalogue knurl

A cataloue_knurl is a type of knurl pattern defined by the user.

```
ENTITY catalogue_knurl
  SUBTYPE OF (knurl);
END_ENTITY;
```

5.2.3 Outer round

Outer_round is a type of turning_feature that is an outline or significant shape that is swept through a complete revolution about an axis. The axis of revolution shall be the same as the Z-axis of the feature.

```
ENTITY outer_round
  ABSTRACT SUPERTYPE OF (ONEOF (outer_diameter, shoulder,step_face))
  SUBTYPE OF (turning_feature);
  diameter: tolerance_length_measure;
END_ENTITY;
```

diameter: The diameter specifies the maximum diametric size of an outer_round feature.

5.2.3.1 Outer diameter

The outer_diameter is a type of outer_round that is a sweeping of an outline specified by a line segment one complete revolution about an axis. The line is finite in length, coplanar with the axis. The outer_diameter may have a constant diameter around the axis of rotation, or it may be tapered.

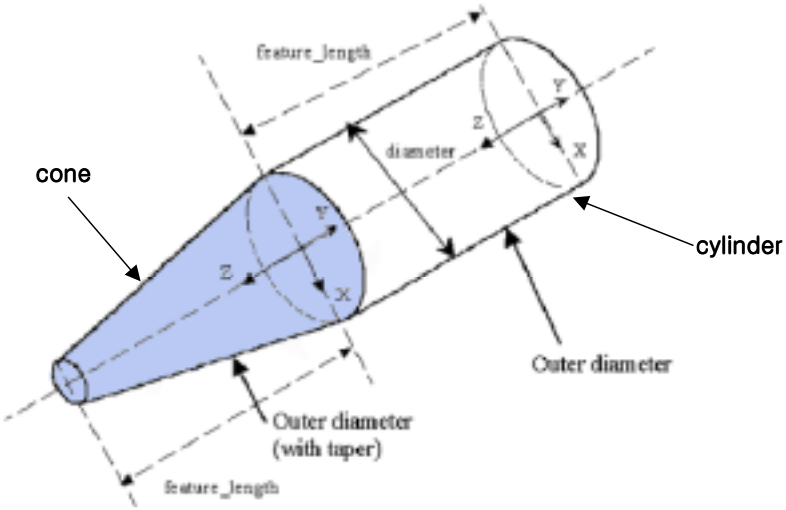


Fig. 5: Outer diameter

```
ENTITY outer_diameter
  ABSTRACT SUPERTYPE OF (ONEOF (cylinder, cone))
  SUBTYPE OF (outer_round);
  feature_length :        toleranced_length_measure;
END_ENTITY;
```

feature_length: The length specifies the size of a outer_diameter feature, measured along the features's axis.

5.2.3.2 Cone

A cone describes a continual transition from one radius to another radius across a certain width.

```
ENTITY cone
  SUBTYPE OF (outer_diameter);
  diameter1 : toleranced_length_measure;
  cone_def :  cone_select_type;
END_ENTITY;
```

Diameter1 : This is the diameter at the side of the feature where the origin of the co-ordinate system of the feature's placement is defined. Its value may be zero if the chuck is on the opposite side.Cone_def : The attribute cone_def makes if possible to select between possibi cone this angle is measured parallel to the z-axis. Its position is at the side of the feature's co-ordinate system. Further details on cone_select_type can be found .

5.2.3.2.1 TYPE cone_select_type

The cone_select_type is used for selecting how to describe a cone. In general, three values are used. Diameter1 and the width of the cone are given in the feature cone. The third parameter may be a second diameter or the angle against the surface normal. For futher details on the definition of a cone .

TYPE cone_select_type = SELECT(cone_diameter2, cone_angle) ;

END_TYPE;

5.2.3.2.2 Cone_diameter2

This entity makes if possible to define the third parameter of a cone as a diameter.

ENTITY cone_diameter2

diamter2 : toleranced_length_measure;

END_ENTITY;

diameter2 : This is the diameter at the opposite side of the feature's placement co-ordinate system. Its value may also be zero if the chuck is not on this side.

5.2.3.2.3 Cone_angle

The third parameter to describe a cone can also be an angle.

Entity cone_angle

angle : plane_angle_measure;

END_ENTITY;

angle : The angle's base is parallel to the z-axis and its position is at the side of the feature's co-ordiate system.

5.2.3.2 Cylinder

A cylinder is used for turning a cylindrical surface along the z axis. It is characterised by its radius and width. The radius is along the width constant.

```
ENTITY cylinder
  SUBTYPE OF (outer_round);
  diameter: tolerance_length_measure;
END_ENTITY;
```

diameter: The diameter specifies the maximum diametric size of an outer_round feature.

5.2.3.3 Shoulder

The shoulder is a type of outer_round that is a sweeping of a shape one complete revolution about an axis. The shape shall be specified by two lines that connect at a point and extend infinitely. The enclosed angle shall be smaller than a straight angle. The intersection of the two lines need not be blended with a radius.

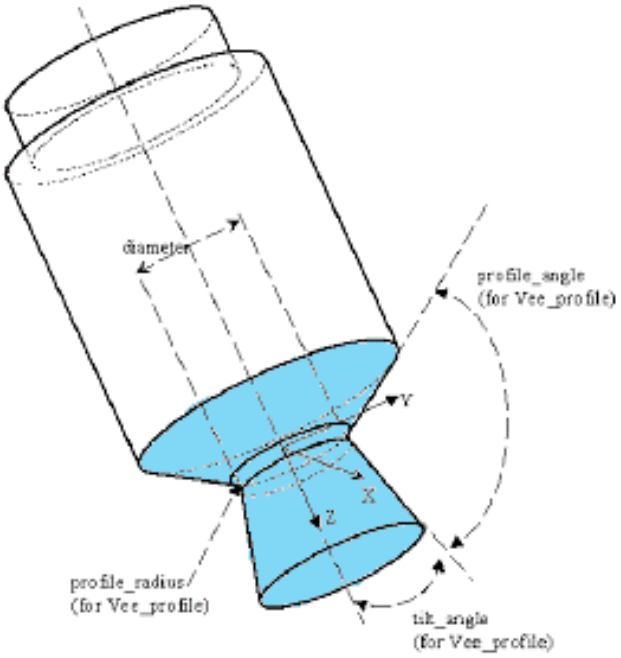


Fig.4: Shoulder

```
ENTITY shoulder
  SUBTYPE OF (outer_round);
  diameter: tolerance_length_measure;
  v_shape_boundary : vee_profile
END_ENTITY;
```

diameter: The diameter specifies the maximum diametric size of an outer_round feature.

v_shape_boundary: This attribute specifies an outline or shape that shall be revolved about an axis.

5.2.4 Step face

A step face is a plane face in radial direction. In contrast to an end face a step face plane is not down to the center, but it has additional information for the transition at its inner diameter. Step faces are used if the radial plane has special requirements on the quality of the surface. For example, when turning groove, both sides of the groove resulting a plane. If its surface has to be finished, a step face feature must be inserted.

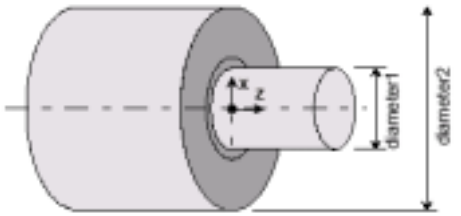


Fig. 5: Step face

ENTITY step_face

SUBTYPE OF (outer_round);

Diameter1 : toleranced_length_measure;

Diameter2 : OPTIONAL toleranced_length_measure;

END_ENTITY;

diameter1: Diameter1 denotes the inner(lower) diameter of the step face. Its value has to be greater than zero.

diameter2: The optional property diameter2 is the outer diameter of the feature. If given this diameter has to be greater than diameter1. If diameter2 is omitted the outer contour of the workpiece is assumed. For this, ths same hintas for the attribute diameter of the end_face is valid.

5.2.5 Revolved feature

A revolved feature is a type of turning_feature that is a sweeping of a planar shape on complete revolution about an axis. The planar shape shall be finite in length, coplanar with the axis of revolution, and shall not intersect the axis of revolution. The axis of revolution shall be the same as the Z-axis of the feature. The revolved_feature may be either an outer shape of a part or a volume removal, depending on the material direction.

```
ENTITY revolved_feature
  ABSTRACT SUPERTYPE OF (ONEOF (revolved_round, revolved_flat, general_revolution,
    groove))
  SUBTYPE OF (turning_feature);
  radius: tolerance_length_measure;
END_ENTITY;
```

radius: The radius specifies the distance from the axis of rotation to define placement of the profile that will be swept about the axis.

5.2.5.1 Revolved round

The revolved_round is a type of revoved_feature that is the sweeping of an arc about an axis.

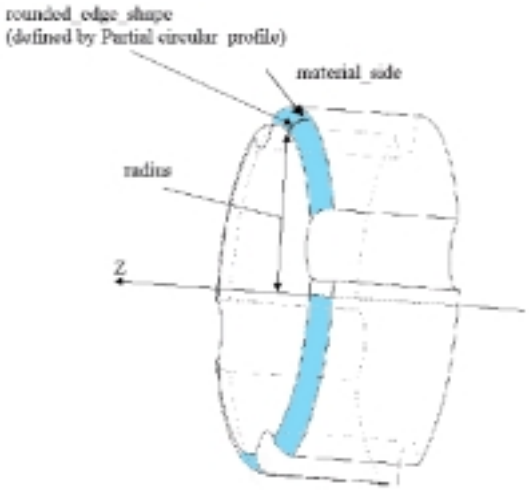


Fig. 6: Revolved round

```
ENTITY revolved_round
  SUBTYPE OF (revolved_feature);
  rounded_edge_shape: partial_circular_profile;
END_ENTITY;
```

rounded_edge_shape: This attribute specifies the arc that when revolved about an axis defines that area on a part for volume removal.

5.2.5.2 Revolved flat

The revolved_flat is a type of revolved_feature that is the sweeping of a straight line about an axis.

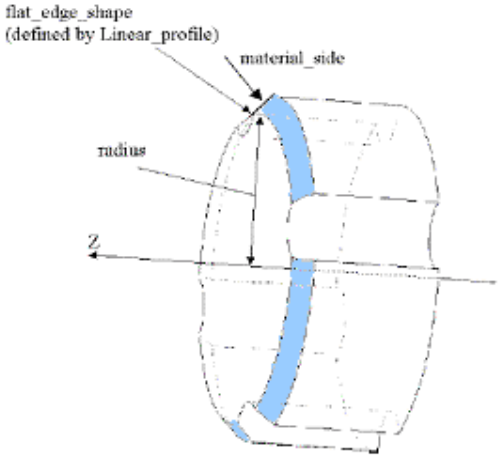


Fig. 7: Revolved flat

```
ENTITY revolved_flat
  SUBTYPE OF (revolved_feature);
  flat_edge_shape: linear_profile;
END_ENTITY;
```

flat_edge_shape: This attribute specifies the line with direction and magnitude that when revolved about an axis defines the area on a part for volume removal.

5.2.5.3 General revolution

The general_revolution is a type of revolved_feature that is an arbitary planar shape swept one complete revolution about an axis. The arbitrary planar shape shall be finite in length, coplanar with the axis of revolution, and shall not intersect the axis of revolution. The general_revolution may be either an outer shape of a part or a volume removal, depending on the material direction.

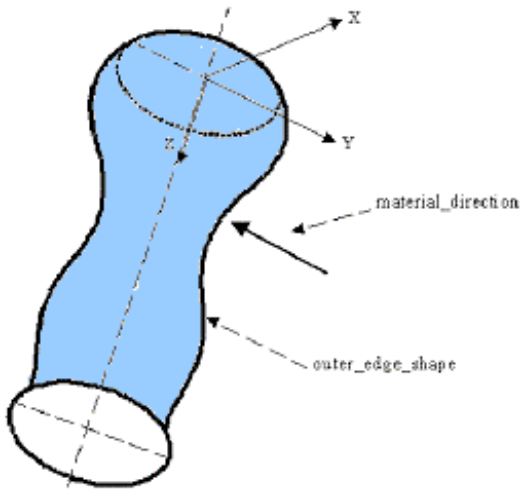


Fig. 8: General revolution

```
ENTITY general_revolution
  SUBTYPE OF (revolved_feature);
  outer_edge_profile: general_profile;
END_ENTITY;
```

outer_edge_profile: The outer_edge_profile specifies an outline or shape that shall be revolved about an axis.

5.2.5.4 Groove

The groove is a type of revolved feature that is a narrow channel or depression that is swept through one complete revolution about an axis.

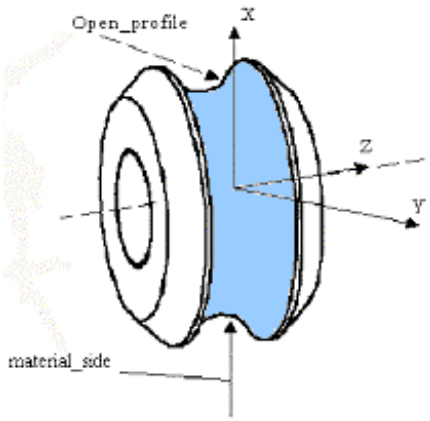


Fig. 9: Groove

```
ENTITY groove
  SUBTYPE OF (revolved_feature);
  sweep: open_profile;
END_ENTITY;
```

sweep: The sweep specifies an outline or shape that shall be revolved about an axis. The general_profile specifies the sweep shape required by a Groove. The placement of the profile shall be along the X-axis of the Groove at a specified distance away from the origin. The orientation of the general_profile is independent of the orientation of the Groove feature. The Groove feature may be defined on different faces of a part depending on the orientation of the profile.

5.2.6 Cut in

A cut in is a kind of slot or groove. The shape of the cut in is identical to the shape of the used tool. For the orientation of the cut in a direction is used. Note that the co-ordinate system of the feature is at ist symmetrical axis.

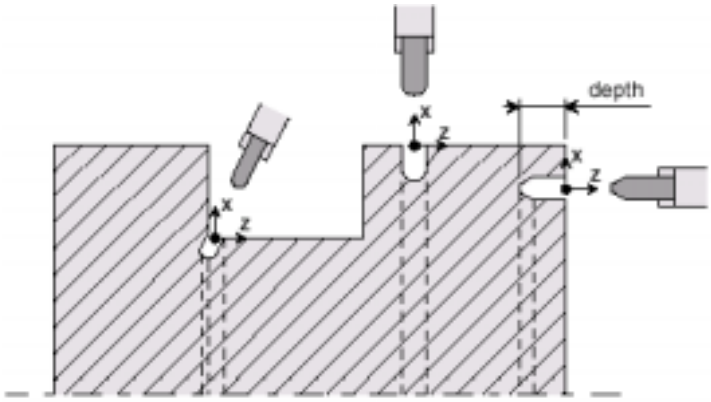


Fig 10: Cut_in

ENTITY cut_in

SUBTYPE OF (turning_feature);

```
    depth : toleranced_length_measure;

    cut_in_direction : OPTINOAL direction;

END_ENTITY;
```

depth: The attribute depth stands for the depth of the cut_in measured from its co-ordinate system in the given direction. By using an appropriate tool(length), it is possible to cut of the end of a bar(shaft), if the z-axis can be reached. For example, if the tool direction is perpendicular to the z-axis and the depth isequal to the radius of the shaft, the end of the shaft is cut off.

cut_in_direction: This optional attribute makes it possible to specify the direction of the cut_in. If omitted, the direction of the tool movement will be perpendicular to the z-axis.

5.2.7 End_face

The end face is used to describe machining of a plane surface on the end wall of a workpiece. The position of the feature is given through the attribute feature_placement of the major class turning_feature. To reach this position material has to be removed. The amount of material is the difference between the raw piece and the desired feature position.

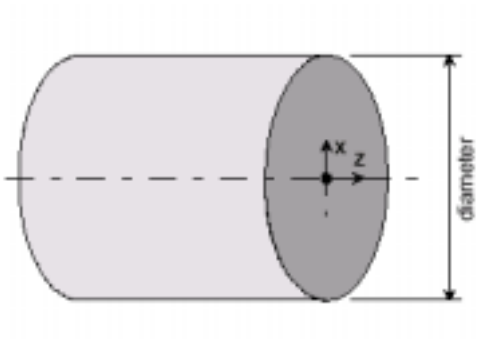


Fig 11: End_face

```
ENTITY end_face

    SUBTYPE OF (turning_feature);

    Diameter : OPTIONAL toleranced_length_measure;

END_ENTITY;
```

diameter: This attribute describe the optional diameter of the end_face. If this value is omitted the outer contour and the real beginning of material is done in the same kind as inside the material. This means that the movement in the air has to be done in the same slow feed rate, which is used for cutting material.

5.3 Machining operations for turning

In this section all machining operations and technology specific data are introduced which will be needed for turning.

5.3.1 Turning technology

This entity defines the technological parameters of the turning operation. It is a subtype of entity *technology* defined in Part 10 of this standard.

```
ENTITY turning_technology
   SUBTYPE OF (technology);
   spindle_speed             : speed_selection_type;
   sync_spindle_and_z_feed   : BOOLEAN;
   inhibit_feedrate_override : BOOLEAN;
   inhibit_spindle_override  : BOOLEAN;
END_ENTITY;
```

spindle_speed : The attribute spindle_speed makes it possible to select between constant spindle speed and constant speed at the surface.

sync_spindle_and_z_feed : If true, feedrate in z and spindle speed are synchronised. Therefore, the pitch of tap can be kept constant at the begin or the end of a thread when feedrate is being accelerated or decelerated.

inhibit_feedrate_override : If true, the feedrate override through the operating panel or by adaptive control systems is not allowed.

inhibit_spindle_override : If true, the spindle speed override through the operating panel or by adaptive control systems is not allowed.

5.3.1.1 Speed select type

The type speed_select_type is used for selection the speed of the lathe' spindle. In general two possibilities have to considered. In the first case a constant spindle speed is given; this speed is measured in revolutions per second. The second case is used for keeping the speed at the surface of the workpiece constant; the unit of this speed is meter per second.

```
TYPE speed_selection_type = SELECT (const_spindle_speed, const_surface_speed);
END_TYPE;
```

5.3.1.2 Const spindle speed

With the entity, a constant rotational speed of the spindle of the lathe can be given. The speed will be kept constant at each position of the diameter.

```
ENTITY const_spindle_speed;
   speed                  : rot_speed_measure;
END_ENTITY;
```

speed : Constant speed of the spindle. As defined for the property rot_speed_measure, positive values indicate rotation in mathematical positive direction of the Z-axis.

5.3.1.3 Const surface speed

If this entity for defining the speed is chosen, the spindle speed will be adapted in the way that the speed at the surface is constant. This means that the smaller the diameter of the machined surface is the higher is the spindle speed and vice versa.

```
ENTITY const_surface_speed;
   speed                : speed_measure;
  max_speed             : OPTIONAL rot_speed_measure;
END_ENTITY;
```

speed : This attribute describes a constant speed at the surface of the workpiece. The speed means the tangential velocity at the tool tip while removing material.

max_speed : With this optional attribute the maximal speed can be limited. While decreasing the diameter of the machined surface, the spindle speed has to be raised. For very small radii the spindle speed becomes very high and is limited by the capabilities of the lathe. If max_speed is given, the speed will be limited as defined by the user. As defined for the property speed_measure, positive values indicate movement in positive direction of the aixs and vice versa.

5.3.2 Turning machine functions

This entity describes the state of various functions of the machine, like coolant, chip removal, etc. to be applied during the time span of an operation. It is a subtype of entity machine_functions defined in Part 10 of this standard.

```
ENTITY turning_machine_functions
   SUBTYPE OF (machine_functions);
   coolant                : BOOLEAN;
   coolant_pressure       : OPTIONAL pressure_measure;
   mist                   : OPTIONAL BOOLEAN;
   through_spindle_coolant : BOOLEAN;
   through_pressure       : OPTIONAL pressure_measure;
   axis_clamping          : LIST [0:?] OF identifier;
   chip_removal           : BOOLEAN;
   oriented_spindle_stop  : OPTIONAL direction;
   its_process_model      : OPTIONAL process_model_list;
   other_functions        : SET [0:?] OF property_parameter;
END_ENTITY;
```

coolant : If true, the coolant is activated.

coolant_pressure : Optional specification of the pressure of the coolant system. Only valid if coolant is true.

mist : If true, activate mist coolant. Default is false. Only valid if coolant is true.

through_spindle_coolant : If true, activate coolant through the spindle. Default is false.

through_pressure : Pressure of coolant through the spindle. The value is only valid if through_spindle_coolant is true.

through_pressure : Describe which axes are to be clamped, e.g. X,Z,C. Note that this information is machine dependent and should be avoided.

chip_removal : If true, activate chip removal.

oriented_spindle_stop : If specified, the spindle will stop in the given direction relative to the machine zero position of C-axis in case a spindle stop occurs during or at the end of the workingstep. This option is especially used for complete machining, i.e. if a milling operation is applied.

its_process_model : Optional information for process control.

other_functions : Optional list of other functions of generic type.

5.3.2.1 Process model list

For each workingstep, one or more process models may be started. These are modules for process control like chatter avoidance, thermal compensation, etc.

```
ENTITY process_model_list;
   its_list: LIST [1:?] OF process_model;
END_ENTITY;
```

its_list : List of process models for the current workingstep

5.3.2.1.1 Process model

This entity is used for the definition of one or more process models in the entity *process_model_list*.

```
ENTITY process_model;
   ini_data_file : label;
   its_type      : label;
END_ENTITY;
```

ini_data_file: A filename including path of the file containing the initialisation data of the process model.

its_type: The type of process model (this can be i.e. chatter avoidance, thermal compensation, ...)

5.3.3 Turning machining strategy

Abstract supertype for the description of the strategy used for creating toolpaths for turning operation.

```
ENTITY turning_machining_strategy
   ABSTRACT SUPERTYPE OF (ONEOF (unidirectional, bidirectional, thread_strategy, ,
    explicit))
   overcut_length        : OPTIONAL length_measure;
   allow_multiple_passes : OPTIONAL BOOLEAN;
   cutting_depth         : OPTIONAL List[0:?] OF length_measure;
```

```
END_ENTITY;
```

overcut_length : The overcut on the open side(s) of the feature. It is not allowed for manufacturing of features which are bounded by material on all sides, i. e. grooves.

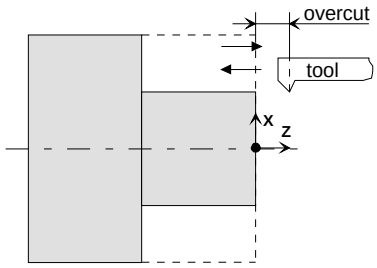


Fig. 12: Overcut of the tool.

allow_multiple_passes : Optional flag only for roughing workingsteps. If true, this is the standard roughing operation with multiple passes, i. e. several layers of material are removed sequentially, taking into account the maximum cutting depth. If false, this is the special roughing operation for pre-cast features with one pass. Default is true.

cutting_depth : The optional *cutting_depth* specifies a list of cutting depth. A cutting depth defines the maximal amount of materal, which can be removed by the tool in one pass. If the cutting depth is smaller than the material to be removed, several layer have to be machined. The cutting depth works in the direction given with the attributes *stepover_direction*. The cutting depth list may be empty. In this cast the selected cutting depth will be implementation dependent. Each list element stands for the cutting depth of the respective pass. If there are more passes than list entries, the last entry is valid for the remaining passes. If there are more list entries than passes, the surplus entries are ignored. The above is valid too, if the controller determines the number of passes.

5.3.3.1 Unidirectional

Turning in a linear fashion, i.e. going from one side to the other, then lifting the tool and going back to the starting point.

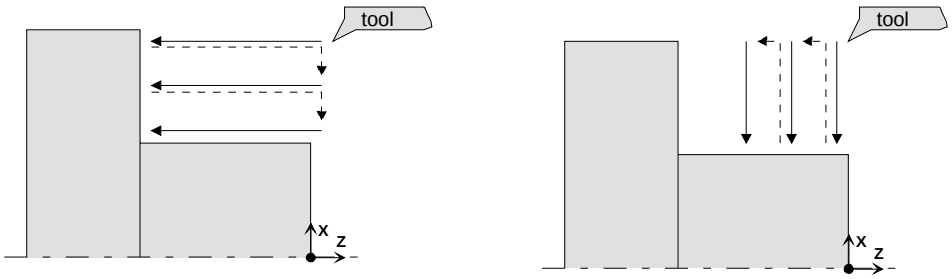


Fig. 13: Unidirectional turning

```
ENTITY unidirectional
   ABSTRACT SUPERTYPE OF (ONEOF (perpendicular_undirectional,
    diagonal_unidirectional, gradual_unidirectional))
   SUBTYPE OF (turning_machining_strategy);
   cutting_direction  : OPTIONAL direction;
   stepover_direction : OPTIONAL direction;
END_ENTITY;
```

cutting_direction: Feed direction of the turning operation, which will remove material. If special attributes like finishing_direction are given, this attribute will be overridden.

second_direction: Stepover direction of the turning operation.

5.3.3.1.1 perpendicular_unidirectional

It is a subtype of entity unidirectional.

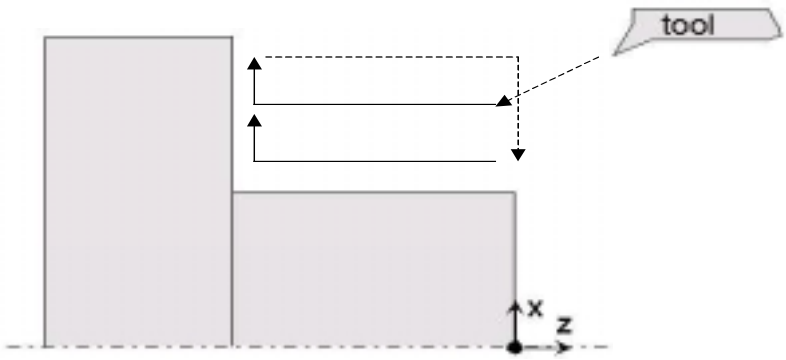


Fig. 14: Perpendicular_unidirectional

```
ENTITY perpendicular_unidirectional
  SUBTYPE OF (unidirectional);
  perpendicular_downward_feed:       OPTIONAL positive_ratio_measure;
END_ENTITY;
```

perpendicular_downward_feed: The attributes specifies change of feedrate of each layer in machining multi-layer.

5.3.3.1.2 Diagonal_unidirectional

It is a subtype of entity unidirectional.

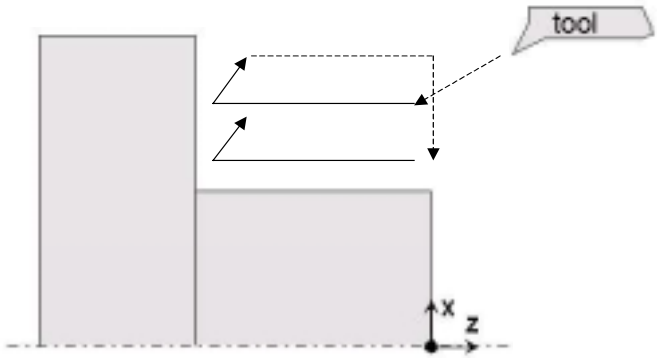


Fig. 15: Diagonal_unidirectional

```
ENTITY diagonal_unidirectional
  SUBTYPE OF (unidirectional);
  pass_return_angle     :       OPTIONAL plain_angle_measure;
  diagonal_downward_feed:       OPTIONAL positive_ratio_measure;
END_ENTITY;
```

pass_return_angle: Diagonally upward cutting angle by each cutting pass in high speed rough-machining cycle. If the value is null, the angle is 45°.

diagonal_downward_feed: The attributes specifies change of feed of each layer in machining multi-layer.

5.3.3.1.3 Gradual_unidirectional

It is a subtype of entity unidirectional.

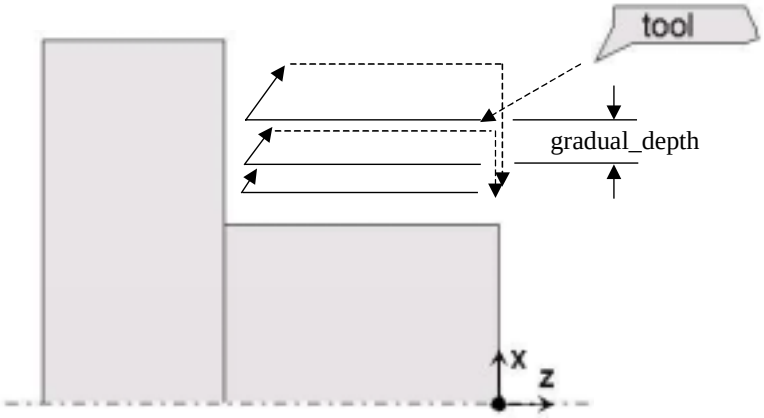


Fig. 16: Gradual

```
ENTITY gradual_unidirectional
  SUBTYPE OF (unidirectional);
  pass_return_angle:          OPTIONAL plain_angle_measure;
  gradual_depth:              OPTIONAL positive_ratio_measure;
  gradual_downward_feed:      OPTIONAL positive_ratio_measure;
END_ENTITY;
```

pass_return_angle: Diagonally upward cutting angle by each cutting pass in high speed rough – machining cycle. If the value is null, the angle is 45°.

gradual_depth: Reduced cutting depth as a percentage of the programmed value.

gradual_downward_feed: Reduced cutting feed as a percentage of the programmed value.

5.3.3.2 Bidirectional

Bidirectional turning means that cutting is started from one side (i.e. the left side). On reaching the other side the cutting tool moves downwards and then back to the first side, while also removing material. Note that special tools are needed for bidirectional turning. These tools must be able to cut their way in the second direction. For example a tool, used for grooving, which blade is along the front side is not able to cut sideways.

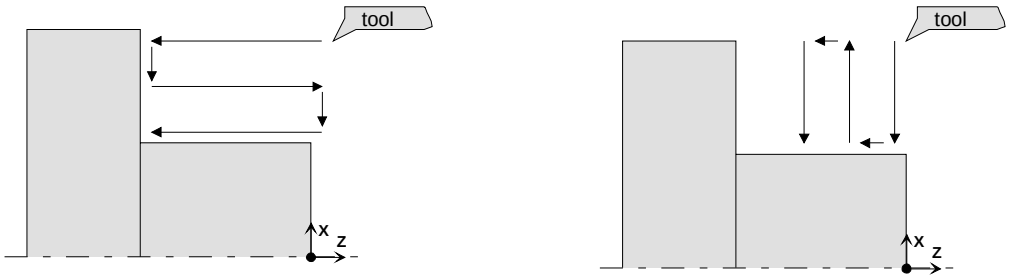


Fig. 17: Bidirectional

```
ENTITY bidirectional
  SUBTYPE OF (turning_machining_strategy);
  perpendicular_downward_feed:       OPTIONAL positive_ratio_measure;
  cutting_direction:        OPTIONAL direction;
END_ENTITY;
```

perpendicular_downward_feed: The attribute specifies feedrate of stepover_direction.

cutting_direction: Feed direction of the first toolpath of the turning operation. If special attributes like finishing_direction are given, this attribute will be overridden.

5.3.3.3 thread_strategy

Thread turning type is a sub class of turning_machining_strategy for cutting a thread.

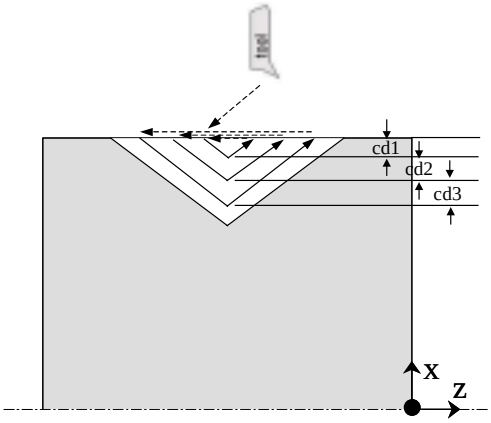


Fig. 18: Thread turning type

```
ENTITY thread_strategy
  SUBTYPE OF (turning_machining_strategy);
  cut_in_amonut_function : user_defined_function;
  threading_direction    : threading_direction_type;
END_ENTITY;
```

cut_in_amount_function: The attribute specifies cut_in_amount.

threading_direction_type: The attribute specifies pass-pattern of machining a thread.

5.3.3.3.1 Threading direction type

This entity describe pass pattern of machining a thread

(a) right **(b) center** **(c) left_zigzag**

Fig. 19: Threading direction type

```
TYPE threading_direction_type = ENUMERATION OF (left, right, left_zigzag,
  right_zigzag, center);
END_TYPE;
```

5.3.3.4 Explicit

Any turning strategy which can not be described using any of the above given strategies can be specified using explicit_turning. In this case, an exact definition of all movements needs to be given in the attribute its_toolpath of the entity workingstep.

```
ENTITY explicit
   SUBTYPE OF (turning_machining_strategy);
END_ENTITY;
```

5.3.4 Turning machining operation

This is the base class of all operations for turning. It includes all necessary attributes to describe technology and cutting strategy. It is a subtype of entity machining_operation defined in Part 10 of this standard.

In general, there are two types of machining operations: roughing and finishing. The roughing is to remove all material from the original raw piece surface down to the bottom or side of the feature minus the finishing allowance in multiple passes. The finishing will then remove the finish allowance to yield the final form of the feature.

```
ENTITY turning_machining_operation
   ABSTRACT SUPERTYPE OF (ONEOF(facing, grooving, contour_turning,
    thread_turning))
   SUBTYPE OF (machining_operation);
   approach : OPTIONAL approach_retract_strategy;
   retract  : OPTIONAL approach_retract_strategy;
    cutting_depth:              OPTIONAL length_measure;
    its_machining_strategy : OPTIONAL turning_machining_strategy;
END_ENTITY;
```

approach : Optional information about approach (plunge) strategy to reach the first cut. If multiple layers have to be cut, as specified by the attribute allow_multiple_passes, this strategy will also be used to move from one layer to the start point of the next layer.

By default, the NC controller determines the approach strategy. It may decide not to use any approach movement at all if the start point of cutting coincides with the end point of cutting for the preceding operation. If its_toolpath is given, this attribute will be ignored.

retract : Optional information about retract strategy after finishing the last cut. By default, the NC controller determines the retract strategy. It may decide not to use any retract movement at all if the end point of cutting coincides with the start point of cutting for the next operation. If its_toolpath is given, this attribute will be ignored.

cutting_depth: The cutting depth in the direction of the tool axis. This can be given to specify a maximal cutting depth smaller than the material removal required by the feature's depth.

Its_machining_strategy: Description of the strategy to be used executing turning operation

5.3.4.1 Approach retract strategy

Base class for the approach (plunge) and retract strategy. All approach and retract strategies are defined relative to the start or end point of the cutting operation, whether this is explicitly given in the operation or determined by the NC controller. The resulting start point of the approach or end point of the retract movement are defined to be the start and end point of the current operation. The feedrate on the approach or retract path is the feedrate specified for the related start or end point, respectively, of cutting.

```
ENTITY approach_retract_strategy
   ABSTRACT SUPERTYPE OF (ONEOF (plunge_strategy, air_strategy, along_path));
END_ENTITY;
```

5.3.4.2 Plunge strategy

This is the base class for all approach movements which include cutting of material. All plunge movements are guaranteed to occur within the boundaries of the underlying feature. All plunge movements will start at the retract plane valid for the current operation. They will end in the start point of the cutting operation, with the tangent of its approach path coinciding with the tangent of the ensuing cutting motion.

```
ENTITY plunge_strategy
   ABSTRACT SUPERTYPE OF (ONEOF (plunge_toolaxis, plunge_ramp))
   SUBTYPE OF (approach_retract_strategy);
END_ENTITY;
```

5.3.4.2.1 Plunge tool axis

Plunge of the tool in the direction of the tool axis.

```
ENTITY plunge_toolaxis
   SUBTYPE OF (plunge_strategy);
END_ENTITY;
```

5.3.4.2.2 Plunge ramp

Plunge on a linear path which forms an angle with the feature surface.

```
ENTITY plunge_ramp
   SUBTYPE OF (plunge_strategy);
   angle : plane_angle_measure;
END_ENTITY;
```

angle : The angle of the ramp movement versus the surface in the end point of the approach.

5.3.4.3 Air strategy

This is the base class for all approach or retract movements through the air. Unlike the *plunge_strategy* types these movements are not limited to the inside of the feature.

```
ENTITY air_strategy
   ABSTRACT SUPERTYPE OF (ONEOF (ap_retract_angle, ap_retract_tangent))
   SUBTYPE OF (approach_retract_strategy);
```

```
END_ENTITY;
```

5.3.4.3.1 Ap retract angle

The movement is heading towards the start or from the end point in an angle to the surface. For cylinder turning, this may typically be an angle of 0 degrees in order to move straight from outside the workpiece into the material.

```
ENTITY ap_retract_angle
   SUBTYPE OF (air_strategy);
   angle         : plane_angle_measure;
   travel_length : length_measure;
END_ENTITY;
```

angle : Approach or lift angle versus the surface in the end point of the approach or the start point of the lift, respectively.

travel_length : The length of the angular approach. After *travel_length* has been reached, the tool will proceed to the retract plane using the shortest connection and vice versa.

5.3.4.3.2 Approach retract tangent

The movement is heading towards the start or from the end point in a curve. The motion start or ends in the retract plane valid for the current operation. If the specified radius for this motion is smaller than the distance to the retract plane as specified in the attribute *retract_plane* of the current operation, the remaining path will be executed in linear motion perpendicular to the retract plane.

```
ENTITY ap_retract_tangent
   SUBTYPE OF (air_strategy);
   radius : length_measure;
END_ENTITY;
```

radius : The radius of the approach or retract movement.

5.3.4.4 Along path

Approach or lift movement on a general path. This should be used if full control of the tool orientation during approach is required or for other special purposes.

```
ENTITY along_path
   SUBTYPE OF (approach_retract_strategy);
   path : toolpath_list;
END_ENTITY;
```

path : Specification of a general path for approach or lift movement. Note that the path is specified in a special co-ordinate system. The origin is the start or end point of the cutting operation, the axes are oriented like the local co-ordinate system of the feature.

5.3.4.5 Facing

This entity describes turning operation for face machining which is used to describe machining of a plane surface on the end wall of a workpiece. The position of the feature is given through the attribute feature_placement

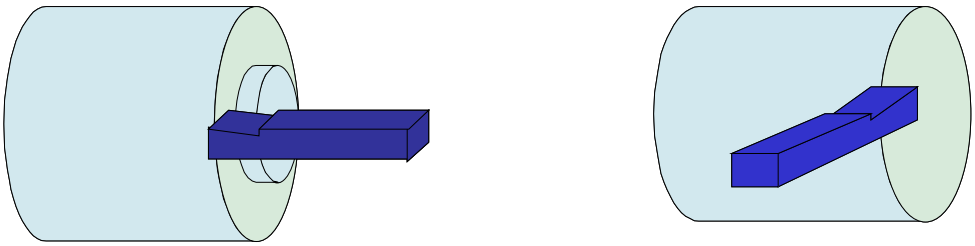


Fig. 20: Facing

```
ENTITY facing
  ABSTRACT SUPERTYPE OF (ONEOF(facing_rough, facing_finish))
  SUBTYPE OF (turning_machining_operation);
  x_relief_amount :           length_measure;
  y_relief_amount :           length_measure;
END_ENTITY;
```

x_relief_amount: The amount of tool relief from walls in the X-axis direction.

z_relief_amount: The amount of tool relief from walls in the Z-axis direction.

5.3.4.5.1 Facing rough

Roughing operation for facing. This operation allows to remove all material until the finishing allowance measured from the features position is reached.

```
ENTITY facing_rough
  SUBTYPE OF (facing);
  finishing_allowance : length_measure;
END_ENTITY;
```

finishing_allowance : The finishing_allowance is a layer of material which will be left on top of the end face. The remaining material is going to be removed by the finishing operation.

5.3.4.5.2 Facing finish

This turning operation is used for finishing for facing. It allows to remove all material until the features position is reached, applying an appropriate strategy to maintain the given tolerances.

```
ENTITY facing_finish
  SUBTYPE OF (facing);
END_ENTITY;
```

5.3.4.6 Grooving

turning operation for groove machining. Depending on special turning tool, it is possible to create a groove. This operation is used for grooving at a certain position without moving sideways.

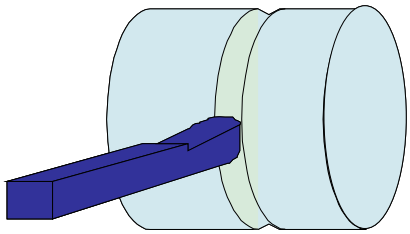


Fig. 21: Grooving

```
ENTITY grooving
  ABSTRACT SUPERTYPE OF (ONEOF(rough_grooving, finish_grooving))
  SUBTYPE OF (turning_machining_operation);
  return_amount:              OPTIONAL length_measure;
  dwell_time_bottom:          OPTIONAL time_measure;
END_ENTITY;
```

return_amount: The amount of tool relief from walls.

dwell_time_bottom: possible dwell time at the bottom of the groove.

5.3.4.6.1 Rough grooving

Roughing operation for grooving. This operation allows to remove all material until the finishing allowance measured from the features position is reached.

```
ENTITY rough_grooving
  SUBTYPE OF (grooving);
  finishing_allowance : length_measure;
END_ENTITY;
```

Finish_allowance: The finishing_allowance is a layer of material which will be left on top of the end face. The remaining material is going to be removed by the finishing operation.

5.3.4.6.2 Finish grooving

This turning operation is used for finishing for grooving. It allows to remove all material until the features position is reached, applying an appropriate strategy to maintain the given tolerances.

```
ENTITY finish_grooving
  SUBTYPE OF (grooving);
END_ENTITY;
```

5.3.4.7 Contour turning

This operation is used for any kind of turning which is not covered by facing or grooving. It has to be distinguished between roughing and finishing operations. Contour turning allows to turn the outer contour of a given feature.

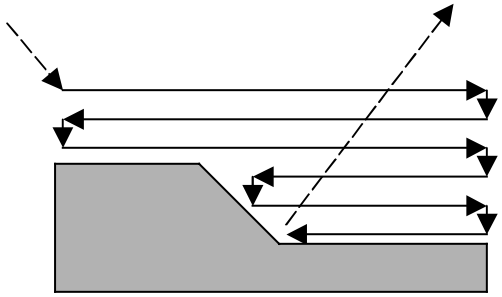


Fig. 22: Contour turning

```
ENTITY contour_turning
   ABSTRACT SUPERTYPE OF (ONEOF(contour_rough_turning, contour_finish_turning))
   SUBTYPE OF (turning_machining_operation);
   x_relief_amount :           length_measure;
   z_relief_amount:           length_measure;
   outer_feedrate :            OPTIONAL length_measure;
END_ENTITY;
```

x_relief_amount: The amount of tool return from walls in the X-axis direction.

z_relief_amount: The amount of tool return from walls in the Z-axis direction.

outer_feedrate: The rate of increase in federate during machining outside the workpiece profile.

5.3.4.7.1 Contour rough turning

Roughing operation for contour turning. This operation allows to remove all material until the finishing allowance is reached.

```
ENTITY contour_rough_turning
   SUBTYPE OF (contour_operation);
   finishing_allowance : length_measure;
END_ENTITY;
```

finishing_allowance : The *finishing_allowance* is a layer of material which will be left on top of the surface. The remaining material is going to be removed by the finishing operation.

5.3.4.7.2 Contour finish turning

This turning operation is used for contour turning. It allows to remove all material until the features position is reached, applying an appropriate strategy to maintain the given tolerances.

```
ENTITY contour_finish_turning
   SUBTYPE OF (contour_operation);
END_ENTITY;
```

5.3.4.8 Thread turning

This entity describe machining a thread.

```
ENTITY thread_turning
  ABSTRACT SUPERTYPE OF (ONEOF(rough_thread_turning, finish_thread_turning))
  SUBTYPE OF (turning_machining_operation);
  relief_amount:              OPTIONAL length_measure;
END_ENTITY;
```

relief_amount: The amount of tool relief from walls.

5.3.4.8.1 rough_thread_turning

Roughing operation for threading. This operation allows to remove all material until the finishing allowance is reached.

```
ENTITY rough_thread_turning
   SUBTYPE OF (thread_turning);
   finishing_allowance : length_measure;
END_ENTITY;
```

finishing_allowance : The *finishing_allowance* is a layer of material which will be left on top of the surface. The remaining material is going to be removed by the finishing operation.

5.3.4.8.2 finish_thread_turning

This finishing operation is used for threading. It allows to remove all material until the features position is reached, applying an appropriate strategy to maintain the given tolerances.

```
ENTITY finish_thread_turning
   SUBTYPE OF (thread_turning);
END_ENTITY;
```

6 Conformance requirements

There is no need for a break down of conformance requirements into classes so far.

Annex A

EXPRESS listing

The following section shows the EXPRESS listing of Part 12: Process Data for Turning.

```
SCHEMA turning_schema;
(* Version -04 date: 01-09-10
 * Author: ISO TC184/SC1/WG7
   Your email contact:  Suk-Hwan Suh (shs@postech.ac.kr) or
                        Heusinger (Stefan.heusinger@isw.uni-stuttgart.de)
 *)
(* ********************************************************** *)
(* Types from machining_schema                    ISO 14649-10   *)
(* ********************************************************** *)

REFERENCE FROM machining_schema(
        axis2_placement_3d,
        bounded_curve,
        cartesian_point,
        direction,
        identifier,
        label,
        length_measure,
        machine_functions,
        machining_operation,
        machining_strategy,
        machining_tool,
        material,
        plane_angle_measure,
        pocket_bottom_condition,
        positive_ratio_measure,
        pressure_measure,
```

```
        property_parameter,
        radial_direction,
        rot_direction,
        rot_speed_measure,
        slot_end_type,
        speed_measure,
        technology,
        time_measure,
        toleranced_length_measure,
        toolpath_list,
        user_defined_function,
        tool_direction_select);

(* ********************************************************** *)
(*   Types defined in process data for turning ISO 14649-12     *)
(* ********************************************************** *)

ENTITY turning_feature
ABSTRACT SUPERTYPE OF (ONEOF(knurl, outer_round, revolved_feature, end_face, cut_in))
SUBTYPE OF (two5D_manufacturing_feature);
material_side : OPTIONAL direction;
END_ENTITY;

(* ********************************************************** *)
(* revolved_feature                                           *)
(* ********************************************************** *)

ENTITY revolved_feature
   ABSTRACT SUPERTYPE OF (ONEOF (revolved_round, revolved_flat, general_revolution, groove))
   SUBTYPE OF (turning_feature);
   radius: toleranced_length_measure;
END_ENTITY;
```

```
ENTITY revolved_round
   SUBTYPE OF (revolved_feature);
   rounded_edge_shape: partial_circular_profile;
END_ENTITY;

ENTITY revolved_flat
   SUBTYPE OF (revolved_feature);
   flat_edge_shape: linear_profile;
END_ENTITY;

ENTITY general_revolution
   SUBTYPE OF (revolved_feature);
   outer_edge_profile: general_profile;
END_ENTITY;

ENTITY groove
   SUBTYPE OF (revolved_feature);
   sweep: open_profile;
END_ENTITY;

(* ********************************************************** *)
(* outer_round                                                *)
(* ********************************************************** *)

ENTITY outer_round
   ABSTRACT SUPERTYPE OF (ONEOF (outer_diameter, shoulder, step_face))
   SUBTYPE OF (turning_feature);
   diameter: tolerance_length_measure;
END_ENTITY;

   ENTITY outer_diameter
      ABSTRACT SUPERTYPE OF (ONEOF (cylinder, cone))
```

```
      SUBTYPE OF (outer_round);

      feature_length :        toleranced_length_measure;

END_ENTITY;



ENTITY cylinder

   SUBTYPE OF (outer_diameter);

   diameter : toleranced_length_measure;

END_ENTITY;



ENTITY cone

   SUBTYPE OF (oute_diameter);

      diameter1 : toleranced_length_measure;
      cone_def :  cone_select_type;
END_ENTITY;

TYPE cone_select_type = SELECT(cone_diameter2, cone_angle) ;

END_TYPE;



ENTITY cone_diameter2

       diamter2 : toleranced_length_measure;

END_ENTITY;



Entity cone_angle

    angle : plane_angle_measure;

END_ENTITY;



ENTITY shoulder

   SUBTYPE OF (outer_round);

   v_shape_boundary : vee_profile;

END_ENTITY;



ENTITY step_face

    SUBTYPE OF (outer_round);
```

```
    Diameter1 : toleranced_length_measure;
    Diameter2 : OPTIONAL toleranced_length_measure;
END_ENTITY;



(* ********************************************************** *)
(* knurl                                                      *)
(* ********************************************************** *)

ENTITY knurl
   ABSTRACT    SUPERTYPE    OF    (ONEOF(straigt_knurl,    diagonal_knurl,    diamond_knurl,
catalogue_knurl))
   SUBTYPE OF (turning_feature);
  base_feature :              turning_feature;
   tooth_depth :           toleranced_length_measure;
    diameter_pitch :        toleranced_length_measure;
    root_fillet:            plane_angle_measure;
    number_of_teeth :       INTEGER;
    major_diameter:         toleranced_length_measure;
    nominal_diameter:       toleranced_length_measure;
END_ENTITY;

ENTITY straigt_knurl
   SUBTYPE OF (knurl);
END_ENTITY;

ENTITY diagonal_knurl
   SUBTYPE OF (knurl);
   helix_angle:    plane_angle_measure;
END_ENTITY;

ENTITY diamond_knurl
   SUBTYPE OF (knurl);
```

```
   helix_angle:    plane_angle_measure;
END_ENTITY;

ENTITY cataloge_knurl
   SUBTYPE OF (knurl);
END_ENTITY;

ENTITY cut_in
    SUBTYPE OF (turning_feature);
    depth : toleranced_length_measure;
    cut_in_direction : OPTINOAL direction;
END_ENTITY;

ENTITY end_face
    SUBTYPE OF (turning_feature);
    Diameter : OPTIONAL toleranced_length_measure;
END_ENTITY;

(* ********************************************************** *)
(* Turning technology                                         *)
(* ********************************************************** *)

ENTITY turning_technology
   SUBTYPE OF (technology);
   spindle_speed             : speed_selection_type;
   sync_spindle_and_z_feed   : BOOLEAN;
   inhibit_feedrate_override : BOOLEAN;
   inhibit_spindle_override  : BOOLEAN;
END_ENTITY;

TYPE speed_selection_type = SELECT (const_spindle_speed, const_surface_speed);
END_TYPE;
```

```
ENTITY const_spindle_speed;
   speed                  : rot_speed_measure;
END_ENTITY;

ENTITY const_surface_speed;
   speed                  : speed_measure;
max_speed               : OPTIONAL rot_speed_measure;
END_ENTITY;


 (* ********************************************************** *)
(* Turning machine functions                                   *)
(* ********************************************************** *)

ENTITY turning_machine_functions
   SUBTYPE OF (machine_functions);
   coolant             :       BOOLEAN;
   coolant_pressure    :       OPTIONAL pressure_measure;
   mist                :       OPTIONAL BOOLEAN;
   through_spindle_coolant:    BOOLEAN;
   through_pressure:           OPTIONAL pressure_measure;
   axis_clamping       :       LIST [0:?] OF identifier;
   chip_removal        :       BOOLEAN;
   oriented_spindle_stop:      OPTIONAL direction;
   its_process_model:          OPTIONAL process_model_list;
   other_functions     :       SET [0:?] OF property_parameter;
END_ENTITY;

ENTITY process_model_list;
   its_list: LIST [1:?] OF process_model;
END_ENTITY;
```

```
ENTITY process_model;
   ini_data_file:  label;
   its_type:       label;
END_ENTITY;

(* ********************************************************** *)
(* Turning type operation                                     *)
(* ********************************************************** *)

ENTITY turning_machining_operation
   ABSTRACT SUPERTYPE OF (ONEOF(facing, grooving, contour_turning,  thread_turning))
   SUBTYPE OF (machining_operation);
   approach:         OPTIONAL approach_retract_strategy;
   retract:          OPTIONAL approach_retract_strategy;
    cutting_depth:        OPTIONAL length_measure;
    its_machining_strategy:  OPTIONAL turning_machining_strategy;
END_ENTITY;

(* ********************************************************** *)
(* approach retract strategy                                  *)
(* ********************************************************** *)
ENTITY approach_retract_strategy
   ABSTRACT SUPERTYPE OF (ONEOF (plunge_strategy,air_strategy,along_path));
END_ENTITY;

ENTITY plunge_strategy
   ABSTRACT SUPERTYPE OF (ONEOF (plunge_toolaxis, plunge_ramp))
   SUBTYPE OF (approach_retract_strategy);
END_ENTITY;

ENTITY plunge_toolaxis
```

```
   SUBTYPE OF (plunge_strategy);
END_ENTITY;

ENTITY plunge_ramp
   SUBTYPE OF (plunge_strategy);
   angle : plane_angle_measure;
END_ENTITY;

ENTITY air_strategy
   ABSTRACT SUPERTYPE OF (ONEOF (ap_retract_angle, ap_retract_tangent))
   SUBTYPE OF (approach_retract_strategy);
END_ENTITY;

ENTITY ap_retract_angle
   SUBTYPE OF (air_strategy);
   angle : plane_angle_measure;
   travel_length : length_measure;
END_ENTITY;

ENTITY ap_retract_tangent
   SUBTYPE OF (air_strategy);
   radius : length_measure;
END_ENTITY;

ENTITY along_path
   SUBTYPE OF (approach_retract_strategy);
   path: toolpath_list;
END_ENTITY;

(* ********************************************************** *)
(* turning operation                                          *)
(* ********************************************************** *)
```

```
ENTITY contour_turning
   ABSTRACT SUPERTYPE OF (ONEOF(contour_rough_turning, contour_finish_turning))
   SUBTYPE OF (turning_machining_operation);
    x_relief_amount:           length_measure;
   z_relief_amount :           length_measure;
   outer_feedrate :            OPTIONAL length_measure;
END_ENTITY;

ENTITY contour_rough_turning
   SUBTYPE OF (contour_turning);
   finishing_allowance: length_measure;
END_ENTITY;

ENTITY contour_finish_turning
   SUBTYPE OF (contour_turning);
END_ENTITY;

ENTITY facing
   ABSTRACT SUPERTYPE OF (ONEOF(facing_rough, facing_finish))
   SUBTYPE OF (turning_machining_operation);
    x_relief_amount:           length_measure;
    z_relief_amount:           length_measure;
END_ENTITY;

ENTITY facing_rough
   SUBTYPE OF (facing);
   finishing_allowance : length_measure;
END_ENTITY;

ENTITY facing_finish
   SUBTYPE OF (facing);
```

```
END_ENTITY;

ENTITY grooving
   ABSTRACT SUPERTYPE OF (ONEOF(rough_grooving, finish_grooving))
   SUBTYPE OF (turning_machining_operation);
    relief_amount:              OPTIONAL length_measure;
   dwell_time_bottom:          OPTIONAL time_measure;
END_ENTITY;

ENTITY rough_grooving
   SUBTYPE OF (grooving);
   finishing_allowance: length_measure;
END_ENTITY;

ENTITY finish_grooving
   SUBTYPE OF (grooving);
END_ENTITY;

ENTITY thread_turning
ABSTRACT SUPERTYPE OF (ONEOF(rough_thread_turning ,  finish_thread_turning))
   SUBTYPE OF (turning_machining_operation);
   relief_amount:              OPTIONAL length_measure;
END_ENTITY;

ENTITY rough_thread_turning
   SUBTYPE OF (thread_turning);
   finishing_allowance: length_measure;
END_ENTITY;

ENTITY finish_thread_turning
   SUBTYPE OF (thread_turning);
END_ENTITY;
```

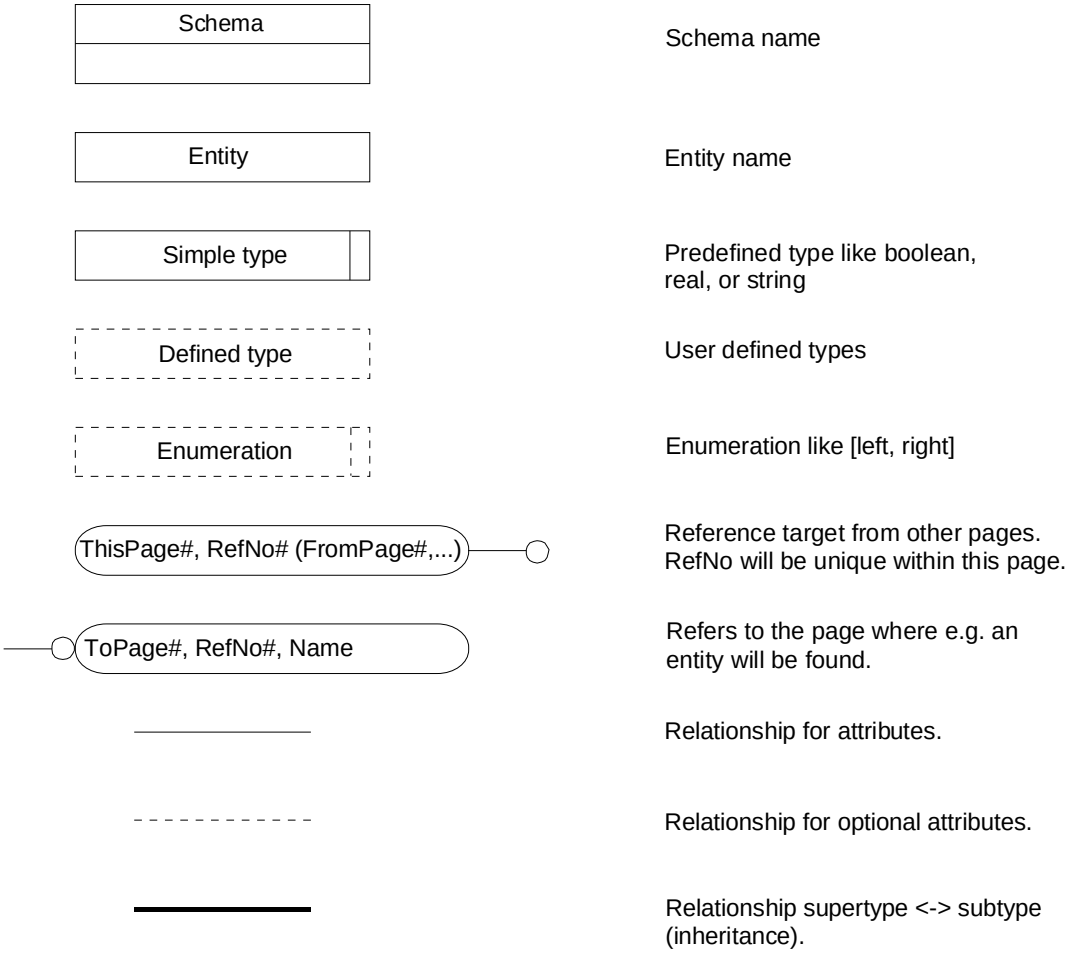
```
(* ********************************************************** *)
(* turning_strategy                                           *)
(* ********************************************************** *)

ENTITY turning_machining_strategy
ABSTRACT SUPERTYPE OF (ONEOF (unidirectional, bidirectional, thread_strategy, explicit));
    allow_multiple_passes:  OPTIONAL BOOLEAN;
   overcut_length :   OPTIONAL length_measure;
   cutting_depth : OPTIONAL LIST [0:?] OF length_measure;
END_ENTITY;

ENTITY unidirectional
   ABSTRACT  SUPERTYPE  OF  (ONEOF  (perpendicular_unidirectional,  diagonal_unidirectional,
gradual_unidirectional))
   SUBTYPE OF (turning_machining_strategy);
   cutting_direction:        OPTIONAL direction;
   stepover_direction:       OPTIONAL direction;
END_ENTITY;

ENTITY perpendicular_unidirectional
   SUBTYPE OF (unidirectional);
   perpendicular_downward_feed:       OPTIONAL positive_ratio_measure;
END_ENTITY;

ENTITY diagonal_unidirectional
   SUBTYPE OF (unidirectional);
   pass_return_angle:                    OPTIONAL plane_angle_measure;
   diagonal_downward_feed:       OPTIONAL positive_ratio_measure;
END_ENTITY;

ENTITY gradual_unidirectional
```

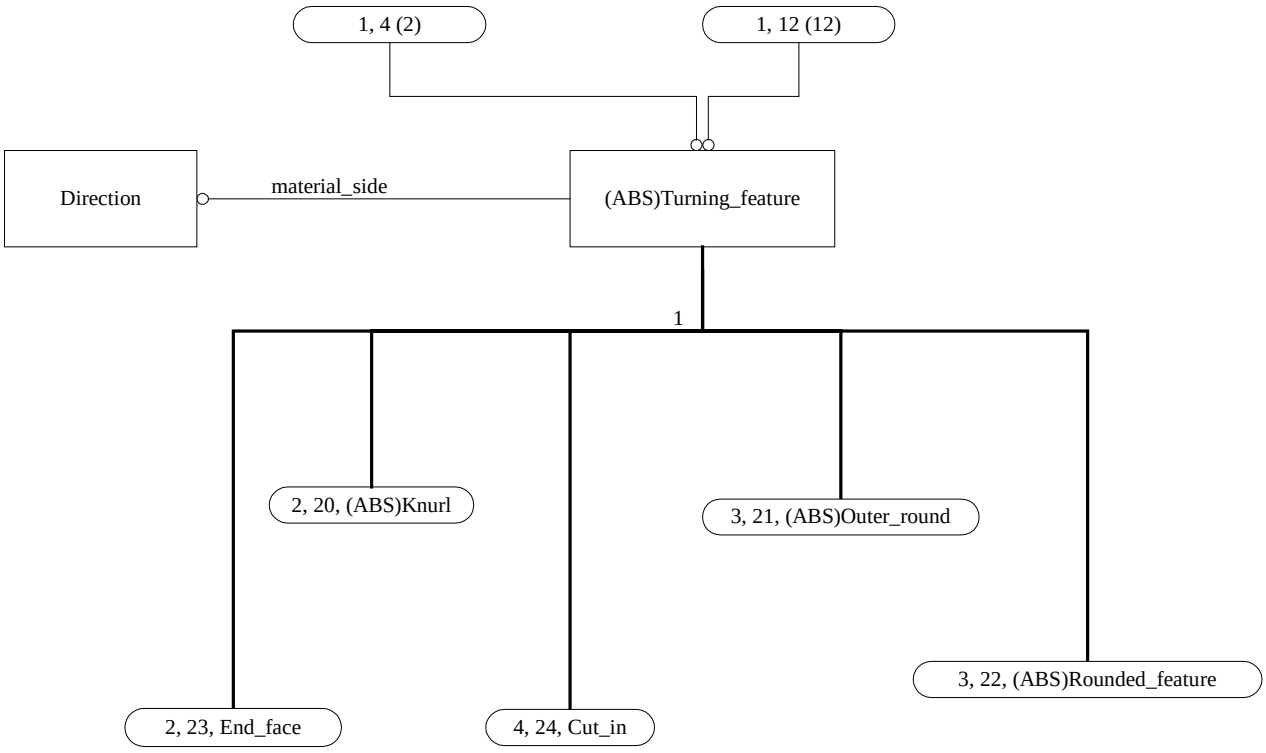
```
   SUBTYPE OF (unidirectional);
   pass_return_angle:          OPTIONAL plane_angle_measure;
   gradual_depth:              OPTIONAL positive_ratio_measure;
   gradual_downward_feed:      OPTIONAL positive_ratio_measure;
END_ENTITY;

ENTITY bidirectional
   SUBTYPE OF (turning_machining_strategy);
   perpendicular_downward_feed:       OPTIONAL positive_ratio_measure;
   cutting_direction:        OPTIONAL direction;
END_ENTITY;

ENTITY thread_strategy
   SUBTYPE OF (turning_machining_strategy);
   threading_direction : threading_direction_type;
   cut_in_amonut_function : user_defined_function;
END_ENTITY;

TYPE threading_direction_type = ENUMERATION OF (left,right,left_zigzag, right_zigzag, center);
END_TYPE;

ENTITY explicit
   SUBTYPE OF (turning_machining_strategy);
END_ENTITY;

END_SCHEMA; (* turning_schema *)
```

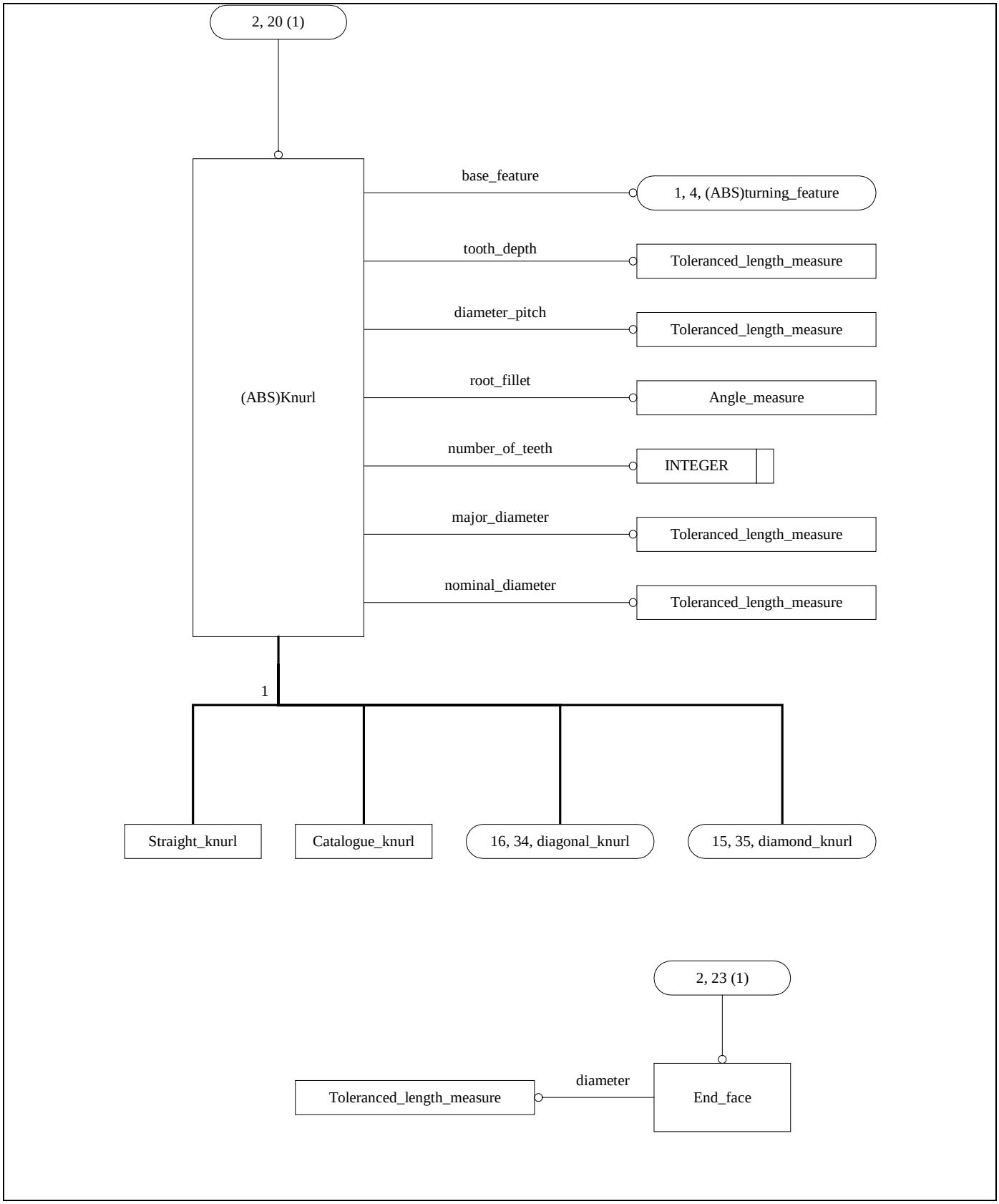
Annex B:

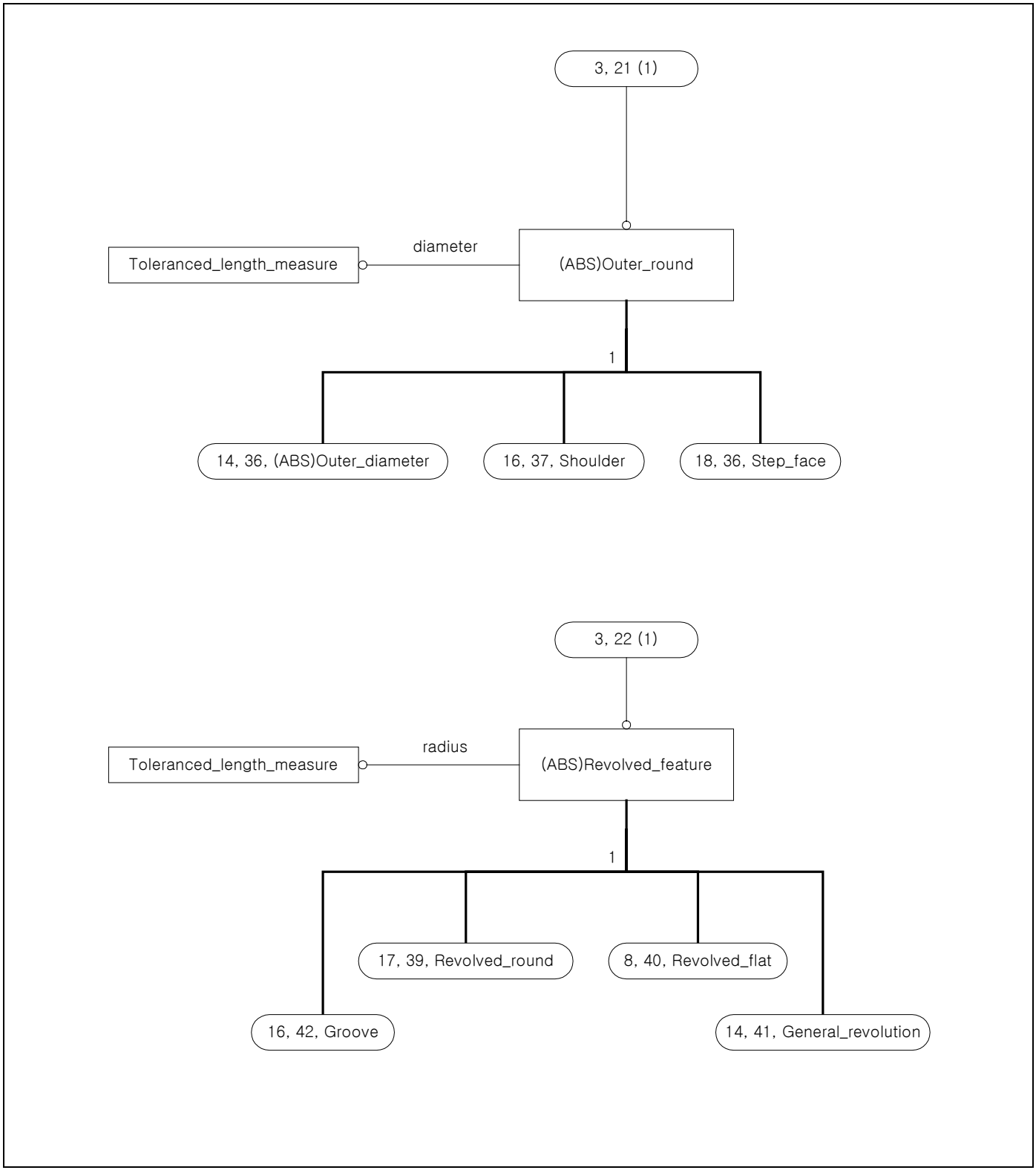
EXPRESS-G

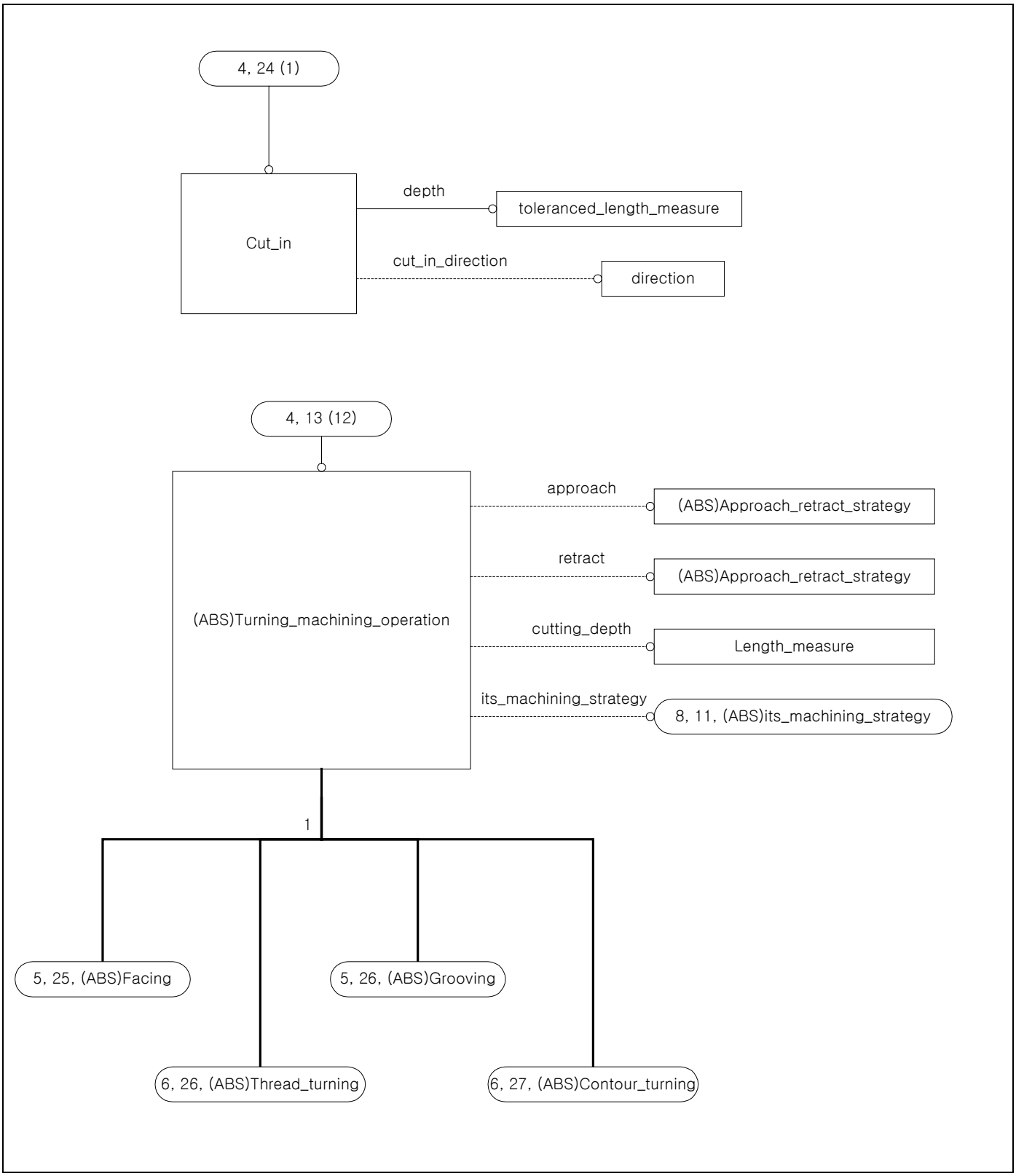
The following section shows the EXPRESS-G figures of Part 12: Process for Turning. According to the notation of EXPRESS-G the used symbols and their respective meaning are listed in brief.

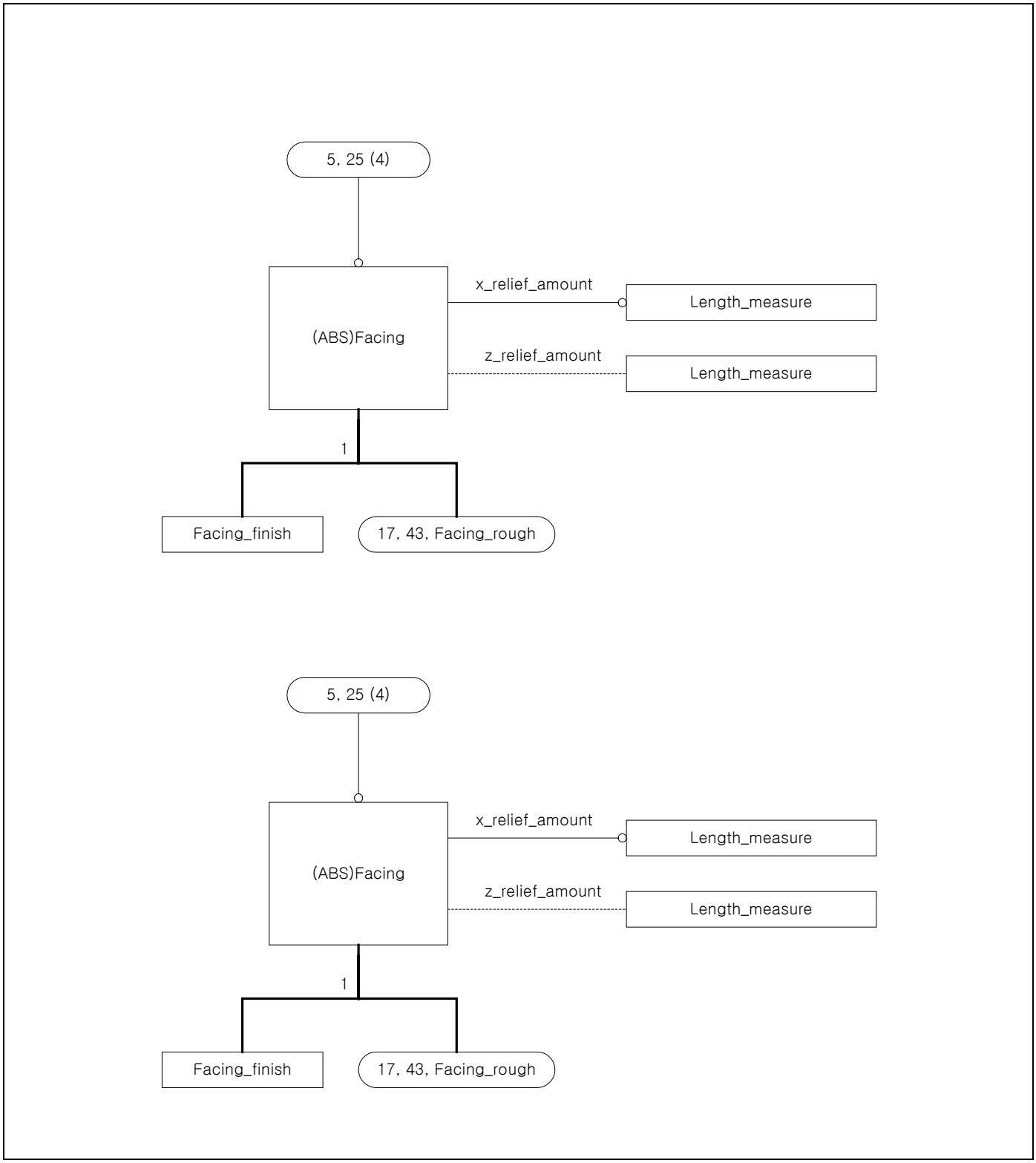




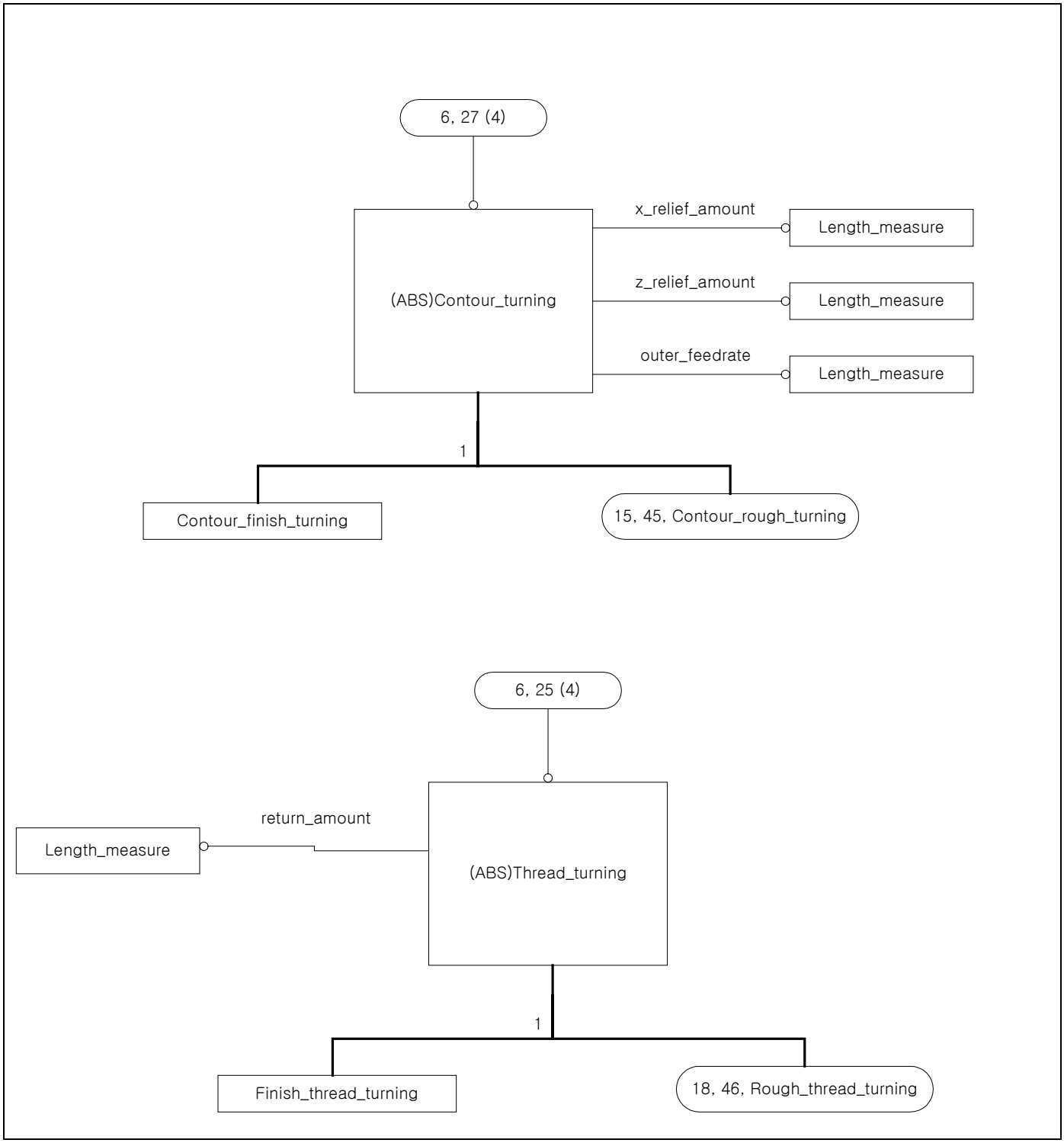




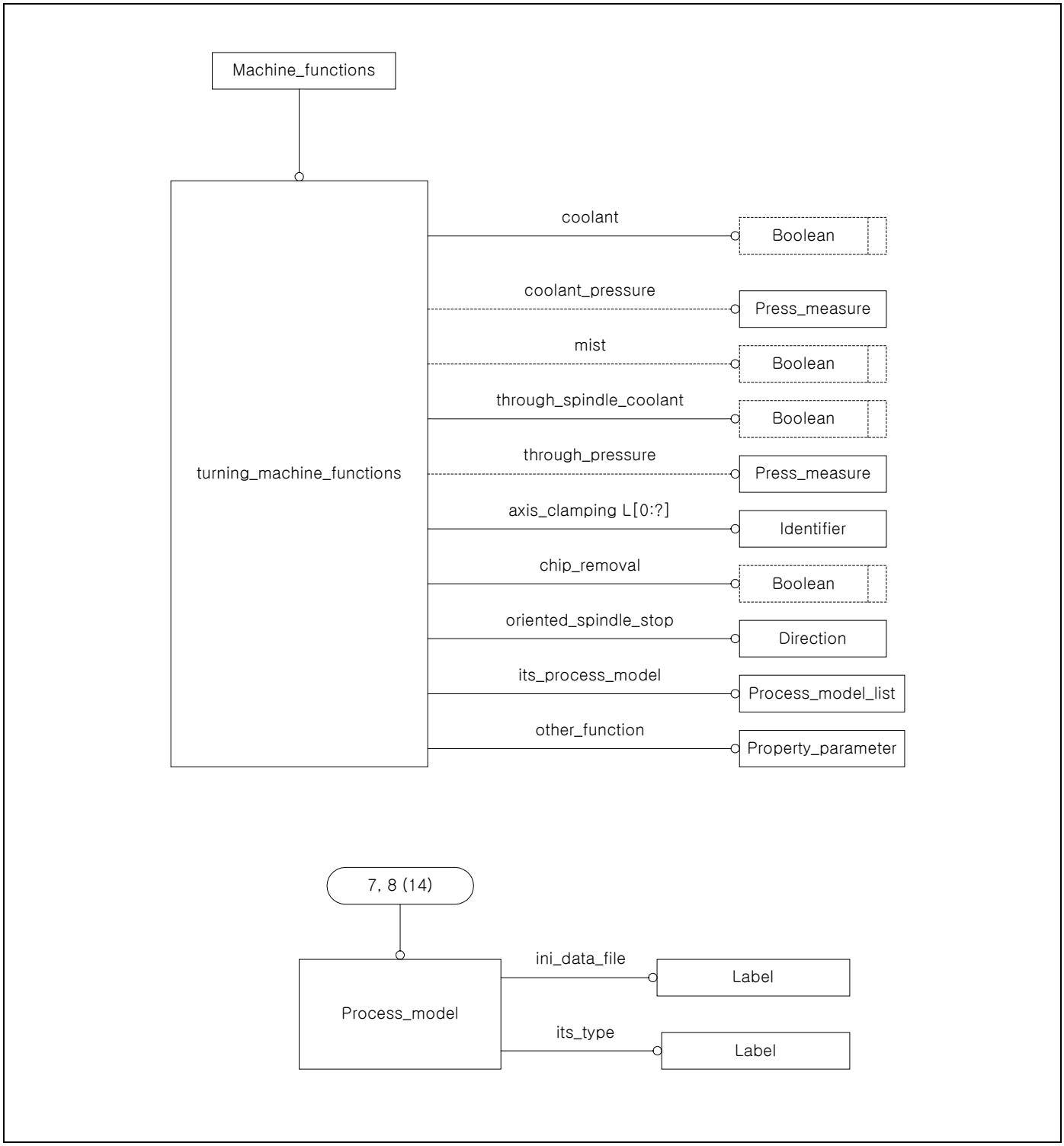


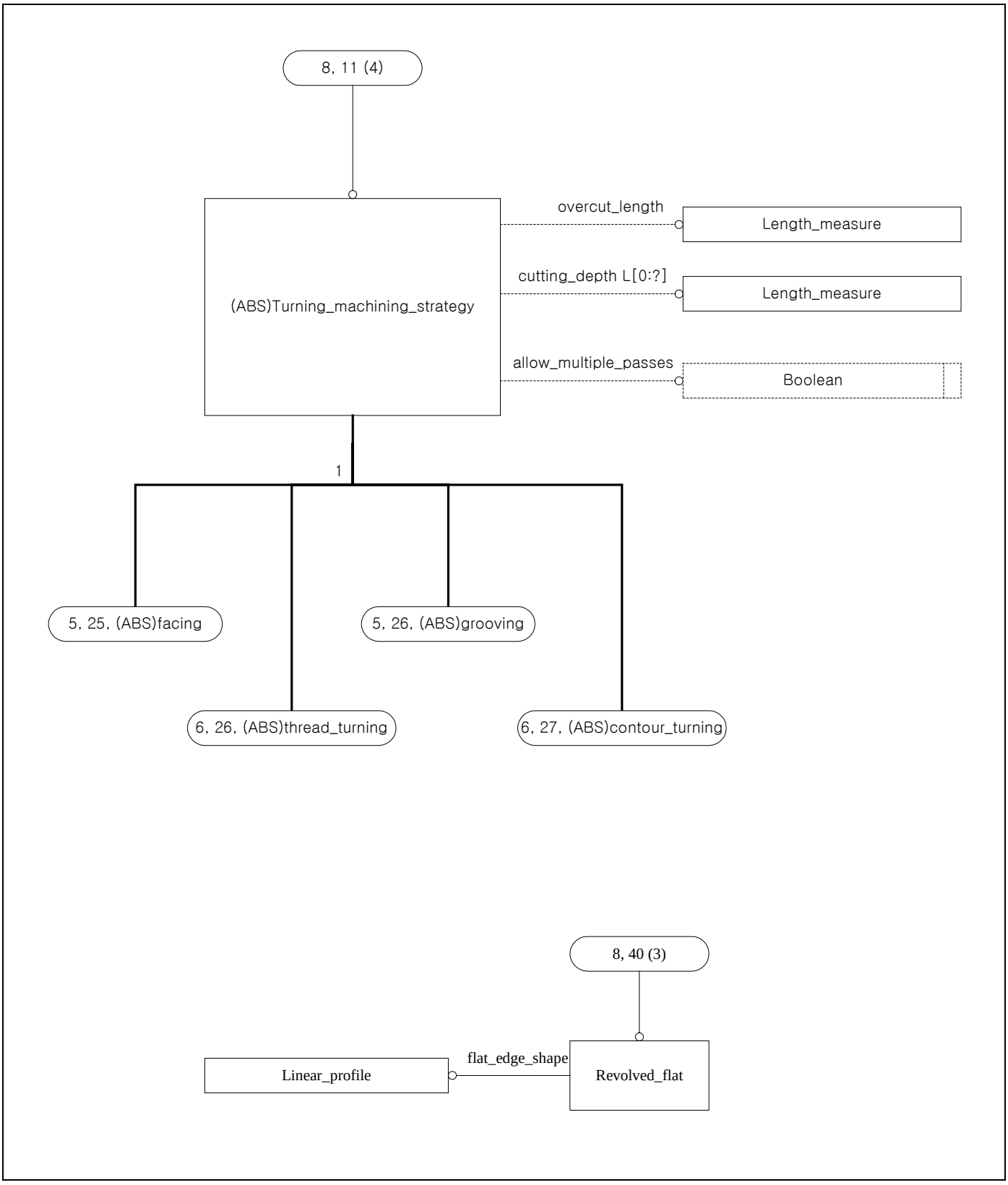


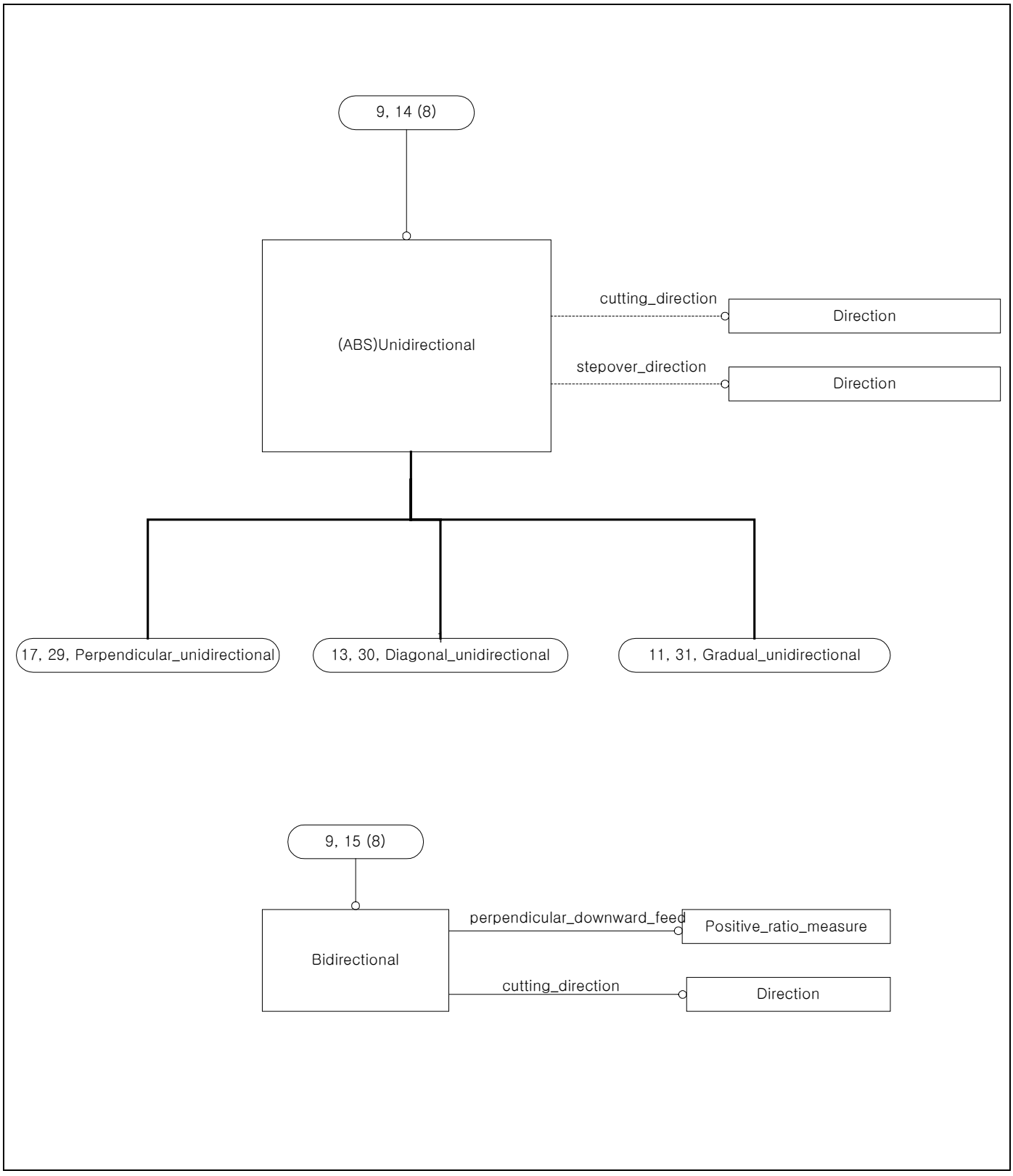
Page B.5

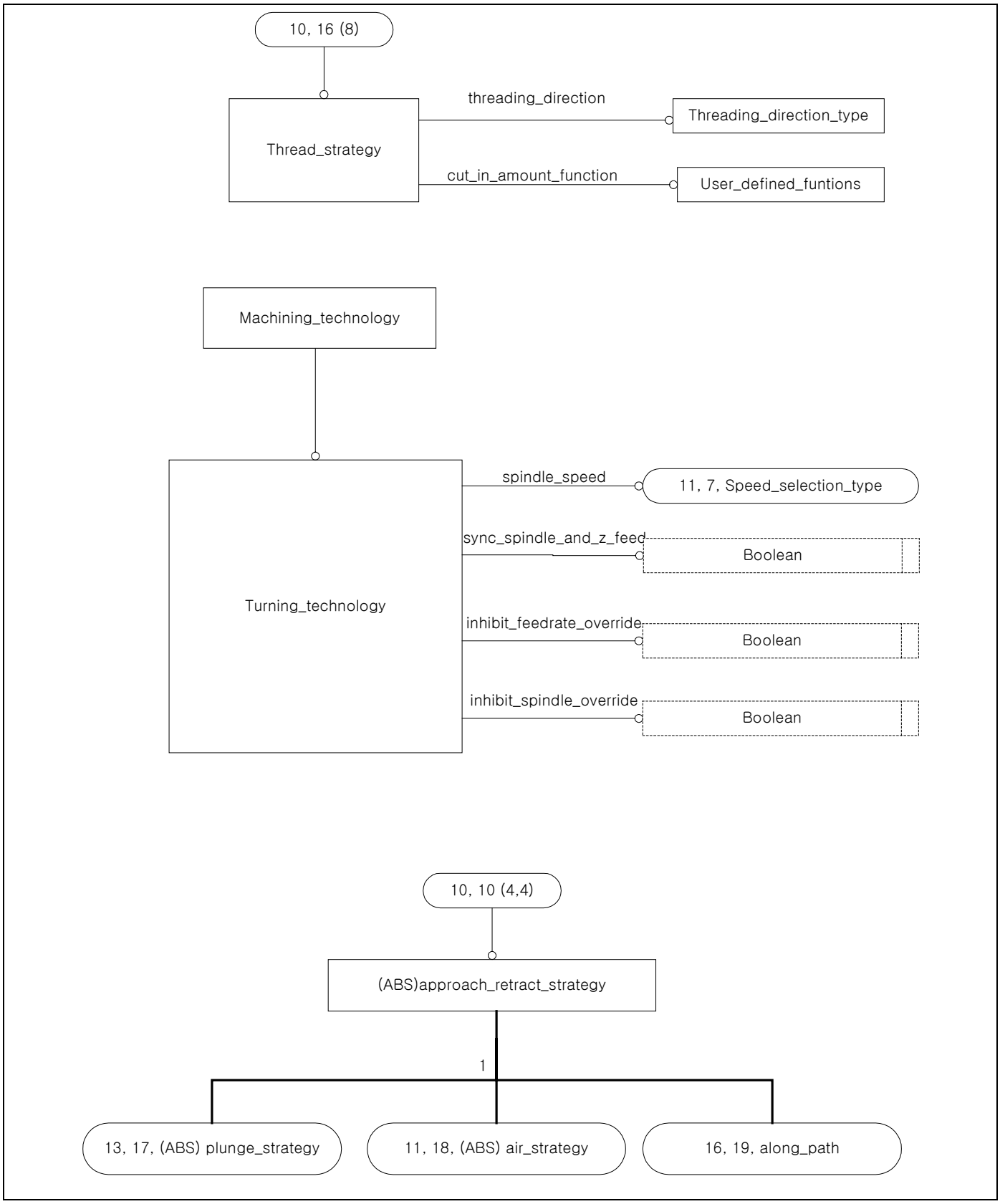


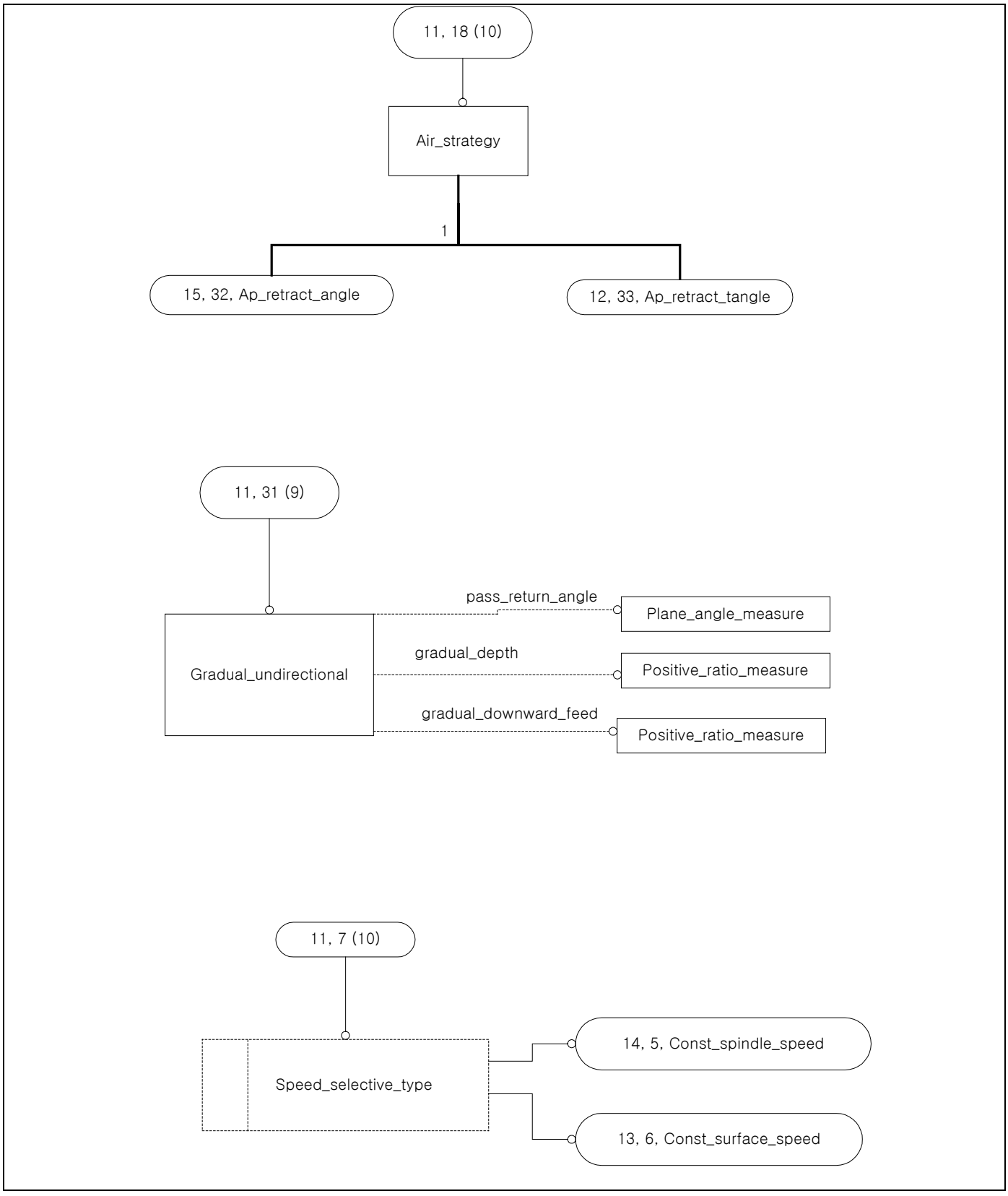
Page B.6

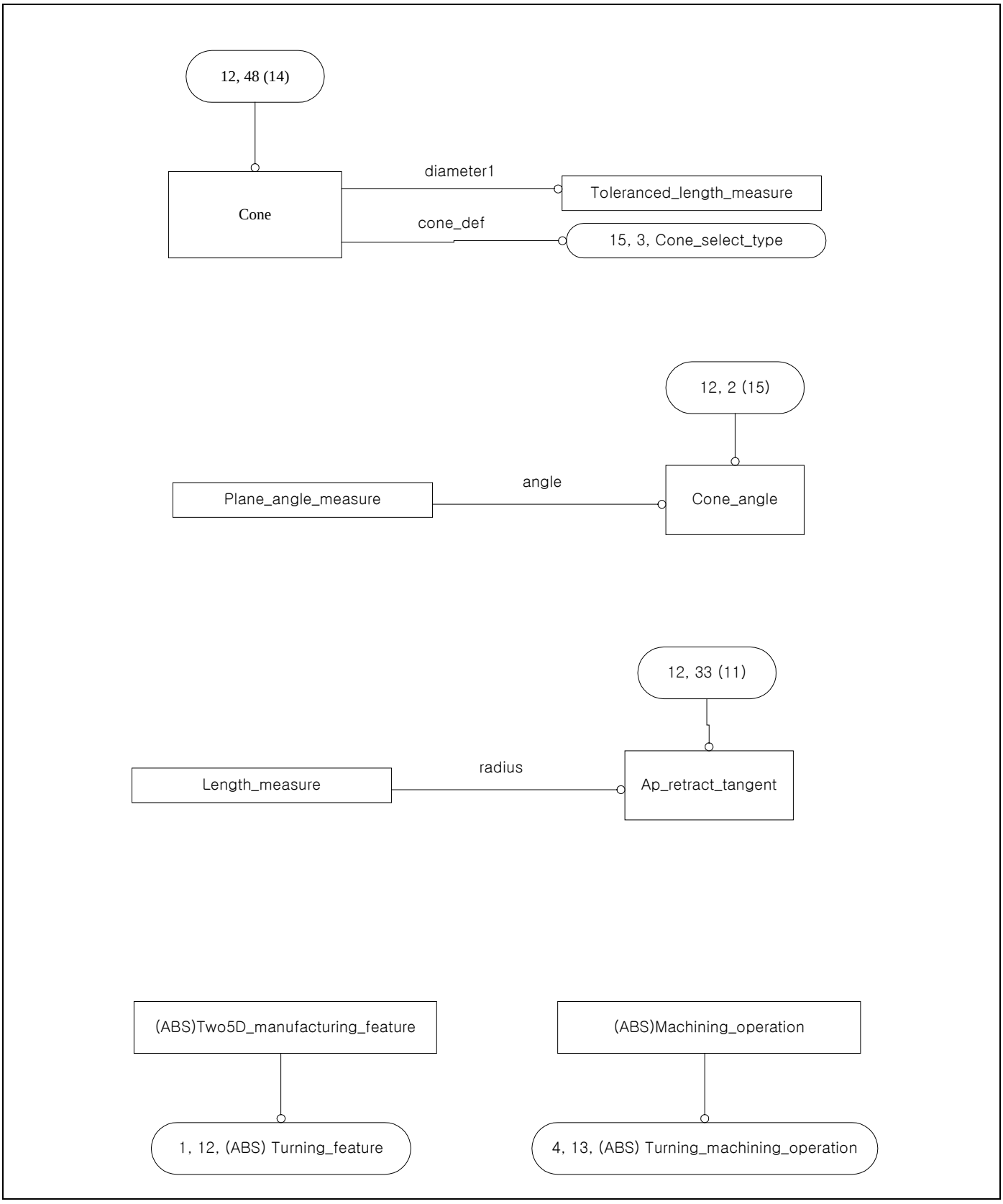


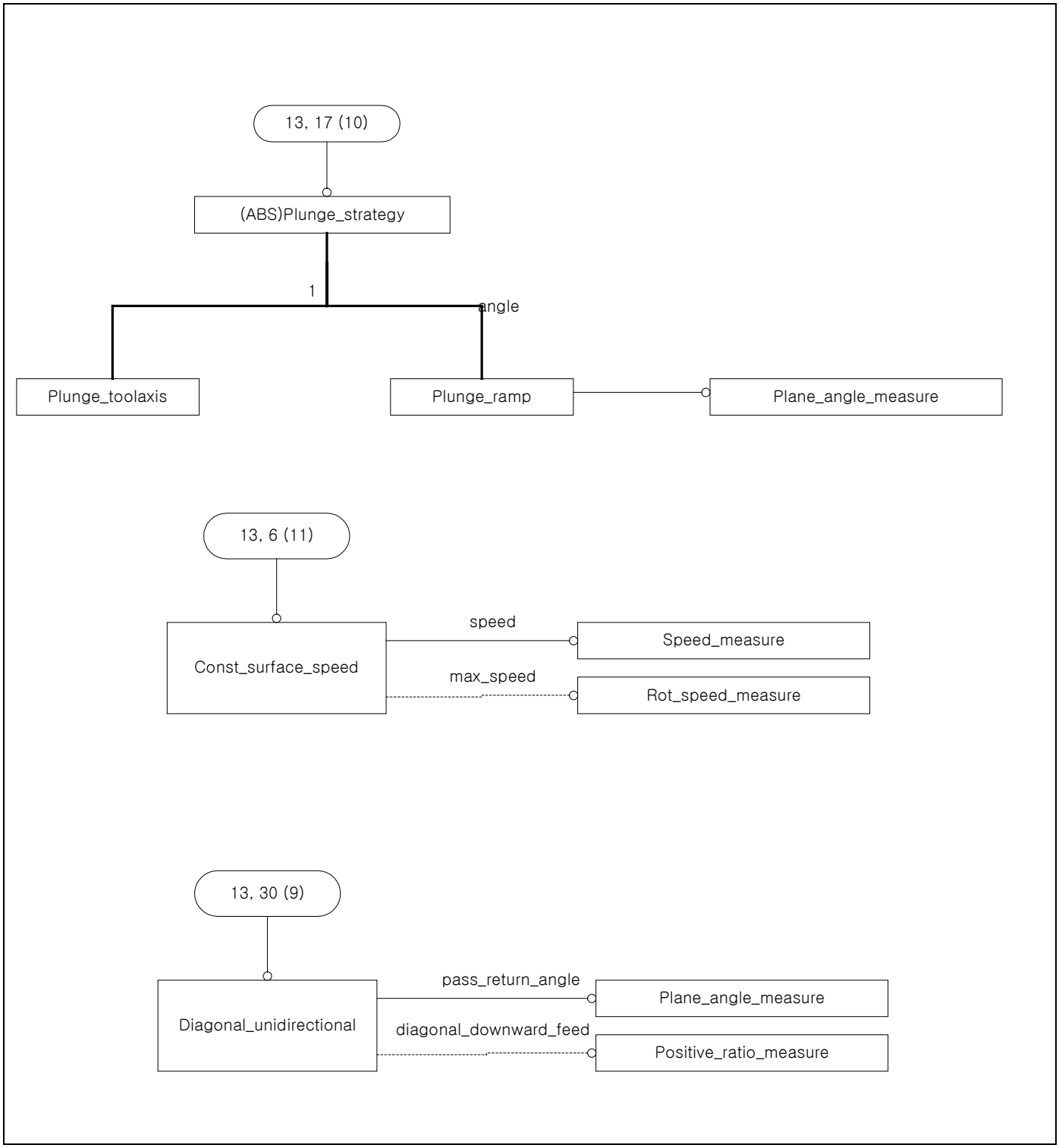


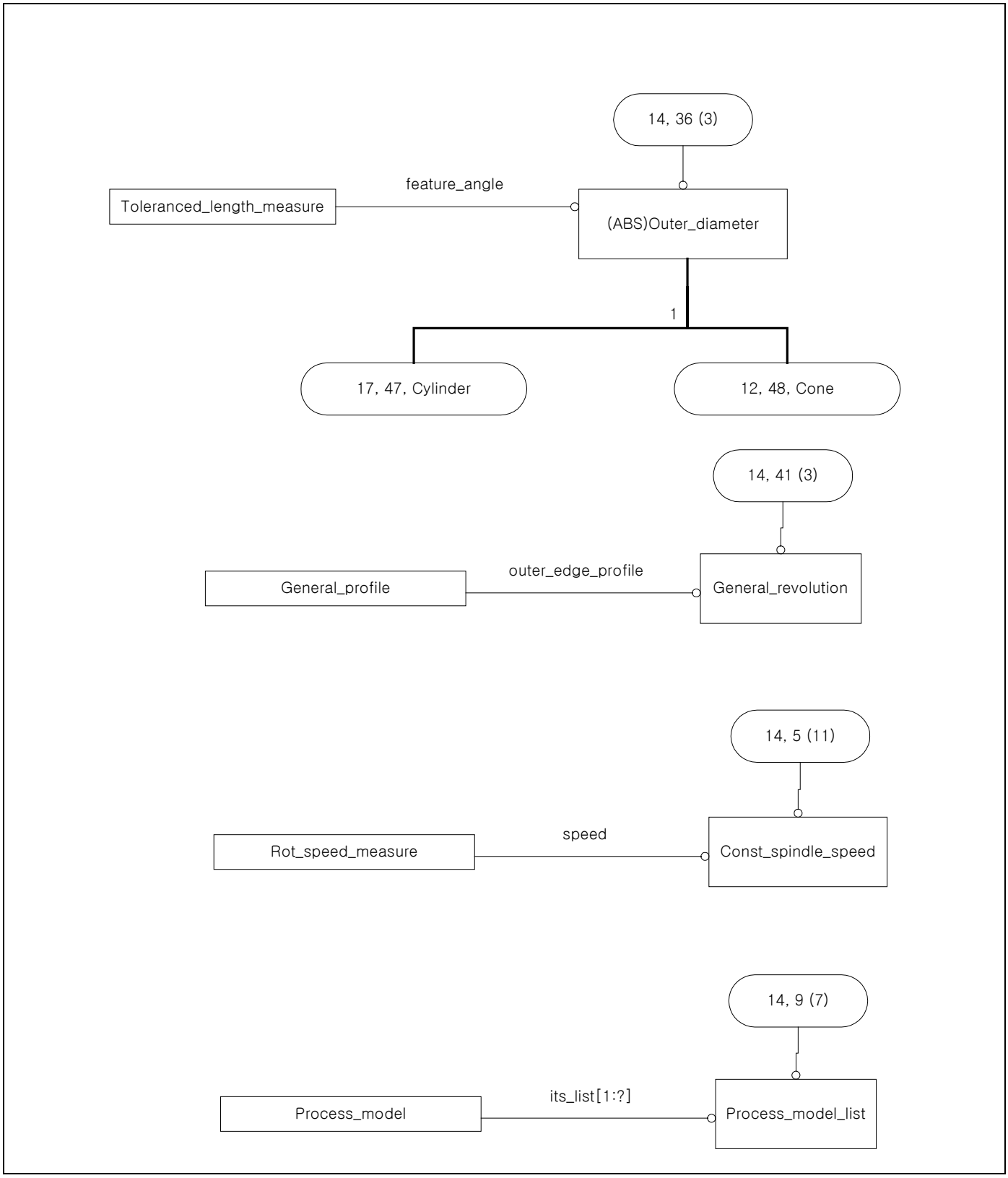


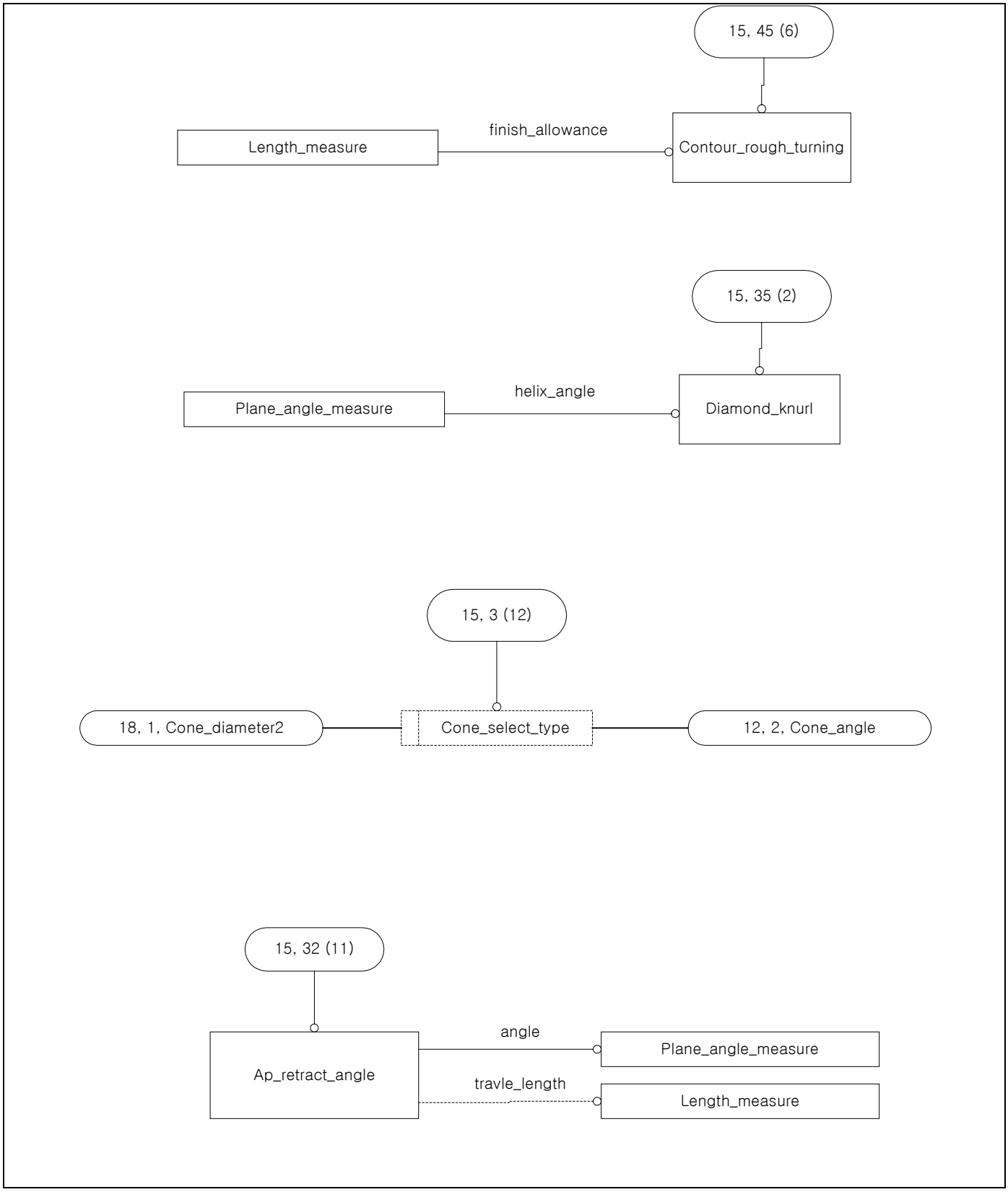


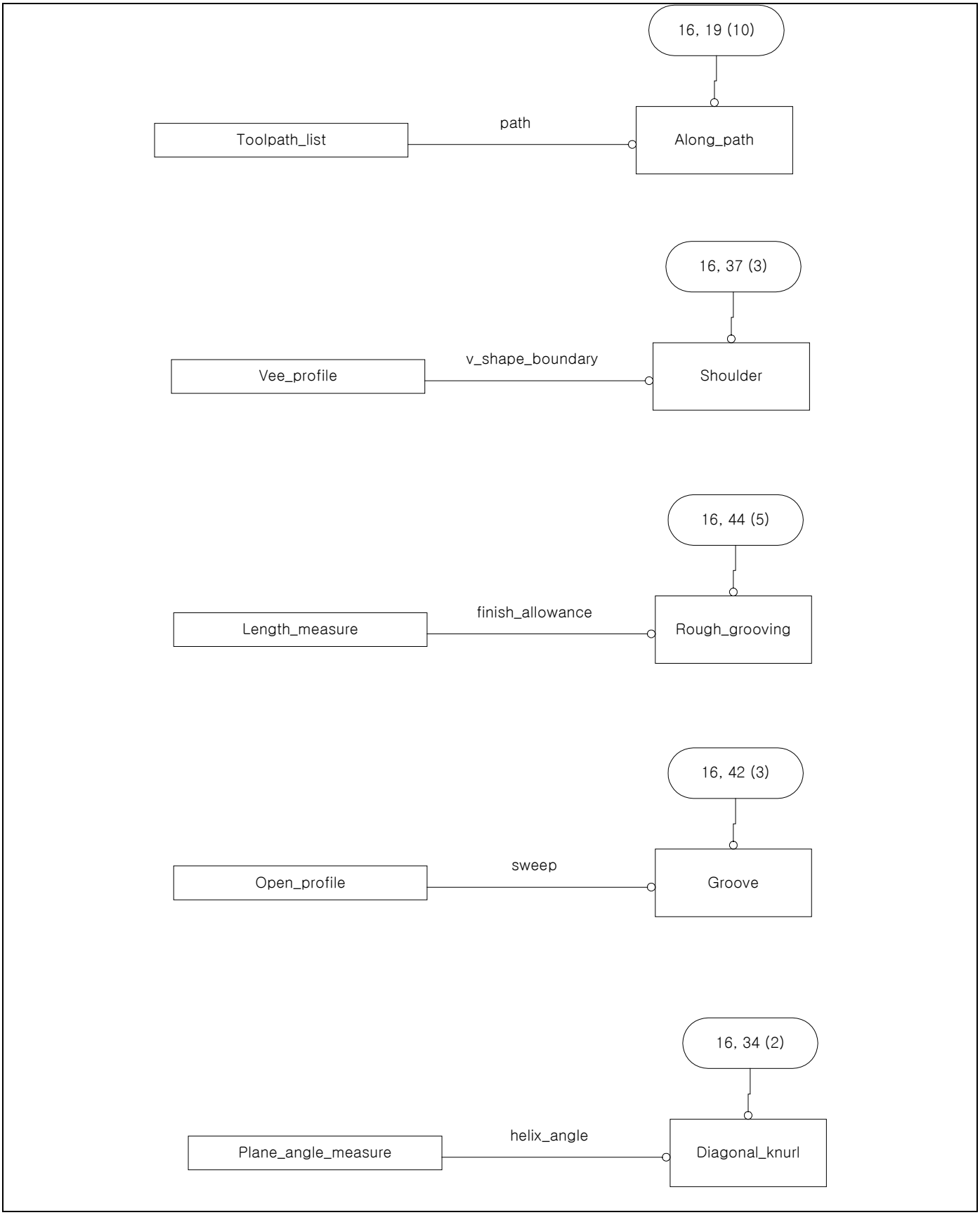


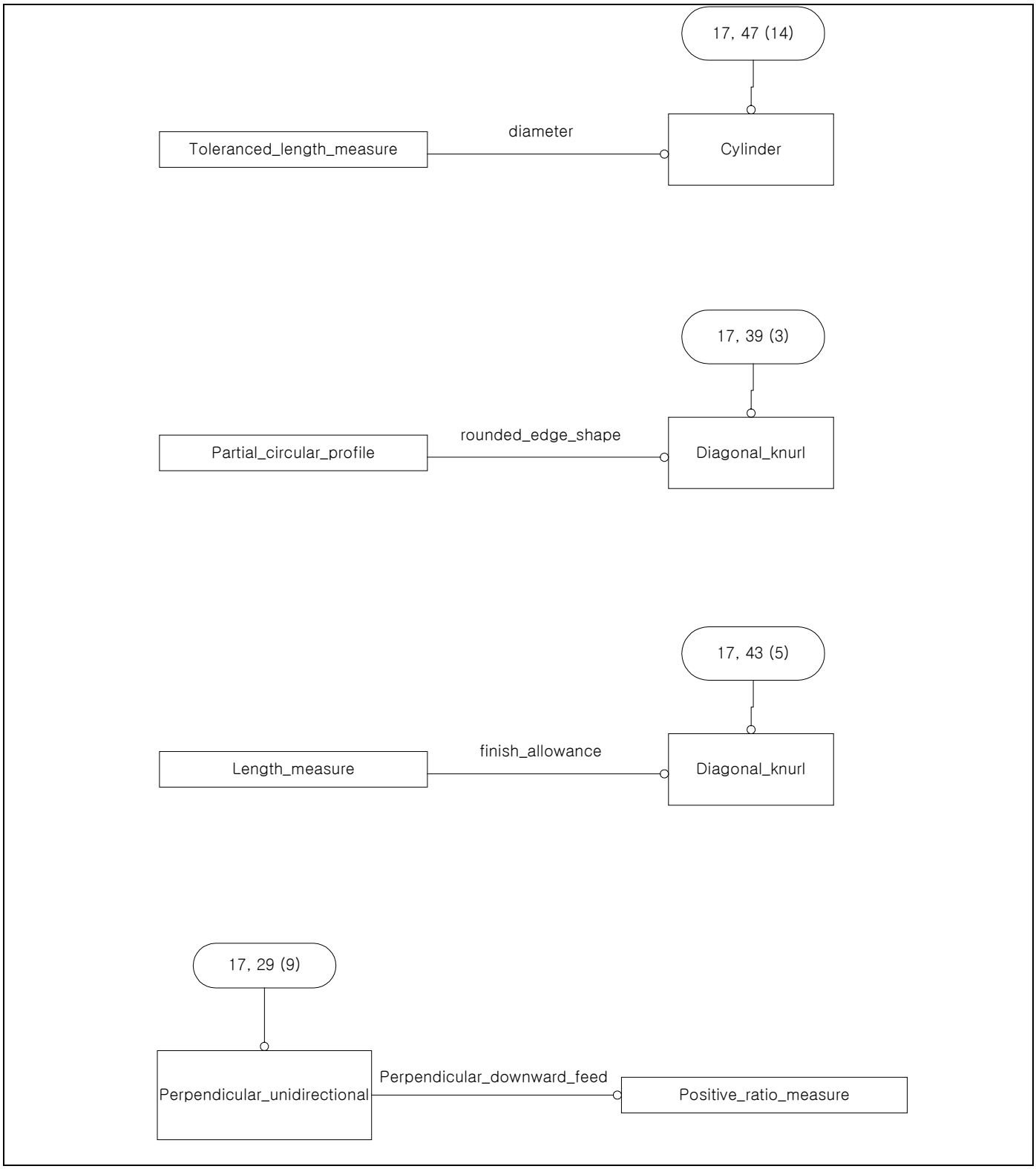


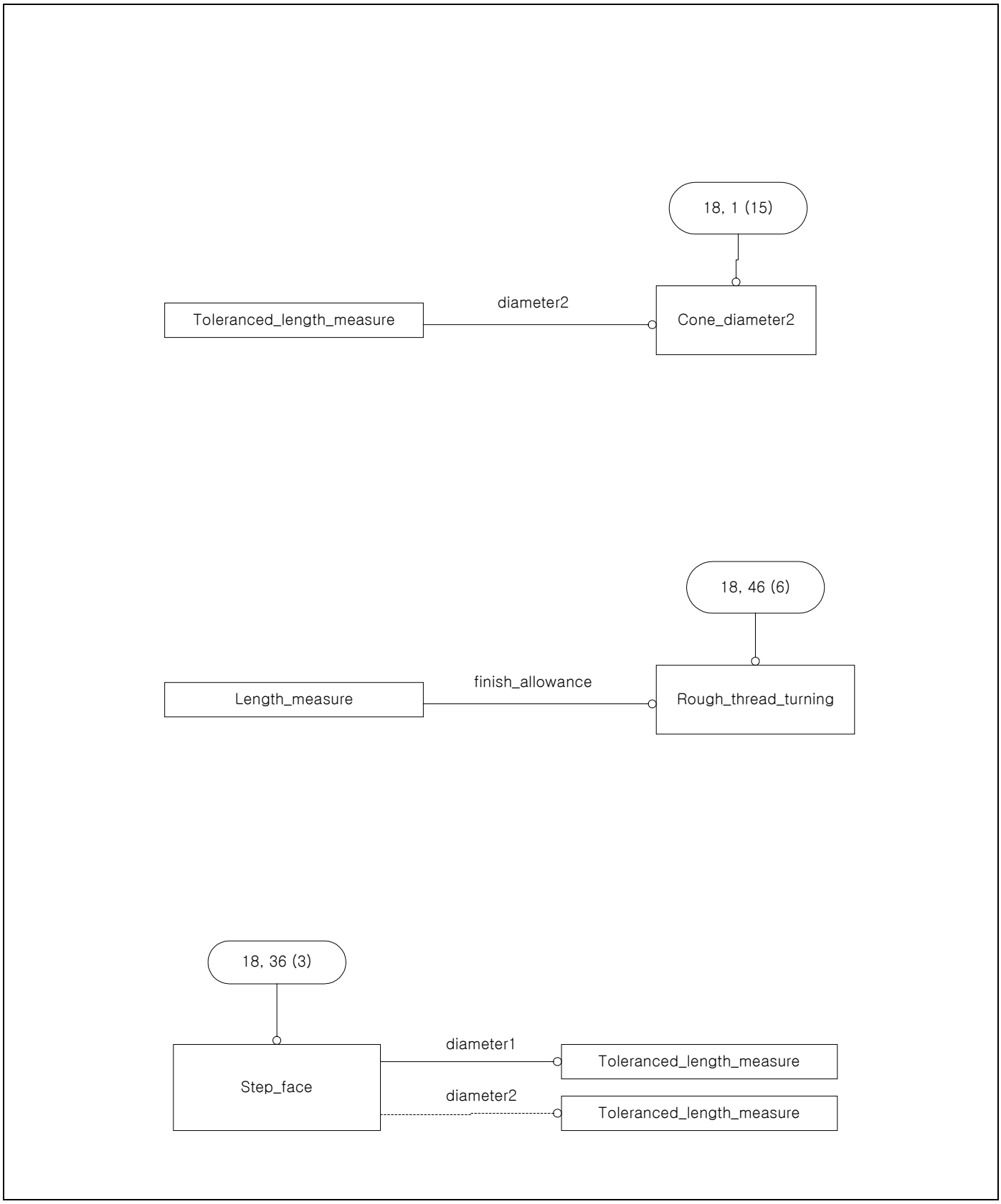












Page B.18