**算子知识图谱设计文档v3.0**

# 算子知识图谱的节点设计

## 1.1 框架节点

### 1.1.1 标签(label)

框架节点的标签命名格式为：“framework”。

### 1.1.2 属性(properties)

框架节点的属性命名包含两个键值对，其中，第一个键名为“name”，值为所采用的深度学习框架；第二个键名为“version”，值为该深度学习框架的版本。示例如下：{name: “Pytorch”, version: “1.5.0”}, {name: “MindSpore”, version: “1.5.0”}, {name: “PaddlePaddle”, version: “1.5.0”}。

## 1.2 类节点

### 1.2.1 标签(label)

类节点的标签命名格式为：“class”。

### 1.2.2 属性(properties)

类节点的属性命名包含两个键值对，其中，第一个键名为“framework”，值为所属的深度学习框架；第二个键名为“name”，值为相应的类名称。示例如下：{framework: “Pytorch”， name: “torch”}, { framework: “MindSpore”， name: “nn”}。

## 1.3 算子节点

### 1.3.1 标签(label)

算子节点的标签命名格式为：“operator”。

### 1.3.2 属性(properties)

算子节点的属性命名包含三个键值对，其中，第一个键名为“framework”，值为所属的深度学习框架；第二个键值对的键名为“name”，值为相应的算子名称；第三个键值对的键名为“full\_name”，值为该算子的完整路径名称。示例如下：{ framework: “Pytorch”， name: “Conv2d”， full\_name：“torch.nn.Conv2d” }。

## 1.4 参数节点

### 1.4.1 标签(label)

参数节点根据该参数是否有多个可选类型可以进一步分为父参数节点和子参数节点：如果该参数只有一种可选类型，则只需构建一个父参数节点；如果该参数有多种可选类型，则需要构建一个父参数节点与若干个连接到父参数节点的子参数节点。其中，父参数节点的标签命名格式为：“parameter”；子参数节点的标签命名格式为“childParameter”。

### 1.4.2 属性(properties)

父参数节点的属性命名包含五个必需键值对和两个非必需键值对，子参数节点的属性在父参数节点属性的基础上还包含一个额外的必需键值对。

父参数节点第一个必需的键值对的键名为“framework”，值为该参数所属的深度学习框架；第二个必需的键值对的键名为“operator”，值为该参数所处的算子名称；第三个必需键值对的键名为“parameter\_order”，值为该参数在所处算子中的顺序序号；第四个必需键值对的键名为“name”，值为相应的参数名称；第五个必需键值对的键名为“dtype”，名为该参数的类型名称。示例如下：{framework：“PyTorch”， operator: “Conv2d”, parameter\_order: 1, name: “in\_channels”, dtype: “int”}。

若该参数有多个可选类型，则构建等同于可选类型数目的子参数节点。其中父参数节点的“dtype”属性的值重定义为参数可选类型的数目，子参数节点在父参数节点属性的基础上再添加一维属性，键名为“dtype\_order”，值为该类型在类型列表中所处的顺序序号。示例如下：若父参数节点有两种可选属性“int”和“tuple”，则父参数节点的属性为{ framework：“PyTorch”， operator: “Conv2d”, parameter\_order: 3, name: “kernel\_size”, dtype: 2}，两个子参数节点的属性分别为{ framework：“PyTorch”， operator: “Conv2d”, dtype\_order: 1, parameter\_order: 3, name: “kernel\_size”, dtype : “int”}, { framework：“PyTorch”， operator: “Conv2d”, dtype\_order: 2, parameter\_order: 3, name: “kernel\_size”, dtype : “tuple”}。

参数节点属性的第一个非必需键值对的键名为“default”，值为该参数的默认值，若该参数没有默认值，则该值为“Null”；第二个非必需键值对的键名为“optional”，值为该参数的可选性，若该参数在相应的算子种是一个可选项，则该值为“True”。示例如下：父参数节点的属性为{ framework：“PyTorch”， operator: “Conv2d”, parameter\_order: 4, name: “stride”, dtype: 2, default: 1, optional: “True”}，两个子参数节点的属性分别为{ framework：“PyTorch”， operator: “Conv2d”, dtype\_order: 1, parameter\_order: 4, name: “stride”, dtype: “int”, default: 1, optional: “True”}，{ framework：“PyTorch”， operator: “Conv2d”, dtype\_order: 2, parameter\_order: 4, name: “stride”, dtype: “tuple”, optional: “True”}。

## 1.5 返回值节点

### 1.5.1 标签(label)

返回值节点根据该返回值是否有多个可选类型可以进一步分为父返回值节点和子返回值节点：如果该返回值只有一种可选类型，则只需构建一个父返回值节点；如果该返回值有多种可选类型，则需要构建一个父返回值节点与若干个连接到父返回值数节点的子参数节点。其中，父返回值节点的标签命名格式为：“return”；子参数节点的标签命名格式为“childReturn”。

### 1.5.2 属性(properties)

返回值节点的属性命名包含五个必需键值对，子返回值节点的属性在父返回值节点属性的基础上还包含一个额外的必需键值对。

父返回值节点第一个必需的键值对的键名为“framework”，值为该返回值所属的深度学习框架；第二个必需的键值对的键名为“operator”，值为该返回值所处的算子名称；第三个必需键值对的键名为“return\_order”，值为该返回值在所处算子中的返回顺序序号；第四个必需键值对的键名为“name”，值为相应的返回值名称【对于一个返回值没给到name情况下，可命名为null标志】；第五个必需键值对的键名为“dtype”，名为该返回值的类型名称。示例如下：{framework：“PyTorch”， operator: “Conv2d”, return\_order: 1, name: “in\_channels”, dtype: “int”}。

若该返回值有多个可选类型，则构建等同于可选类型数目的子返回值节点。其中父返回值节点的“dtype”属性的值重定义为返回值可选类型的数目，子返回值节点在父参数节点属性的基础上再添加一维属性，键名为“dtype\_order”，值为该类型在类型列表中所处的顺序序号。示例如下：若父返回值节点有两种可选属性“int”和“tuple”，则父返回值节点的属性为{ framework：“PyTorch”， operator: “Conv2d”, return\_order: 3, name: “kernel\_size”, dtype: 2}，两个子返回值节点的属性分别为{ framework：“PyTorch”， operator: “Conv2d”, dtype\_order: 1, return\_order: 3, name: “kernel\_size”, dtype : “int”}, { framework：“PyTorch”， operator: “Conv2d”, dtype\_order: 2, return \_order: 3, name: “kernel\_size”, dtype : “tuple”}。

# 算子知识图谱框架内的关系设计

## 2.1 框架节点与类节点之间的关系

### 2.1.1 类型(type)

框架节点与类节点之间关系的类型命名格式为：“classOfFramework”。

### 2.1.2 属性(properties)

框架节点与算子类节点之间关系的属性命名包含一个键值对，其中键名为“name”，值为所连接的类节点的名称。示例如下：{name: “torch”}。

## 2.2类节点与类节点之间的关系

### 2.2.1 类型(type)

类节点与类节点之间关系的类型命名格式为：“subClassOfClass”。

### 2.2.2 属性(properties)

类节点与类节点之间关系的属性命名包含一个键值对，其中键名为“name”，值为所连接的类节点的名称。示例如下：{name: “nn”}。

## 2.3类节点与算子节点之间的关系

### 2.3.1 类型(type)

类节点与算子节点之间关系的类型命名格式为：“operatorOfClass”。

### 2.3.2 属性(properties)

类节点与算子节点之间关系的属性命名包含一个键值对，其中键名为“name”，值为所连接的算子节点的名称。示例如下：{name: “Conv2d”}。

## 2.4算子节点与父参数节点之间的关系

### 2.4.1 类型(type)

算子节点与父参数节点之间关系的类型命名格式为：“parameterOfOperator”。

### 2.4.2 属性(properties)

算子节点与参数节点之间关系的属性命名包含两个键值对，其中第一个键名为“parameter\_order”，值为所连接的参数节点在当前算子的参数列表中所处的顺序；第二个键名为“name”，值为所连接的参数节点的名称。示例如下：{parameter\_order: 1, name: “in\_channels”}。

## 2.5父参数节点与子参数节点之间的关系

### 2.4.1 类型(type)

父参数节点与子参数节点之间关系的类型命名格式为：“oneOfParameter”。

### 2.4.2 属性(properties)

参数节点与子参数节点之间关系的属性命名包含两个键值对，第一个键名为“dtype\_order”，值为所连接的子参数节点在当前参数节点的类型列表中所处的顺序；第二个键名为“dtype”，值为所连接的子参数节点的类型。示例如下：{dtype\_order: 1, dtype: “int”}。

## 2.6算子节点与父返回值节点之间的关系

### 2.4.1 类型(type)

算子节点与父返回值节点之间关系的类型命名格式为：“returnOfOperator”。

### 2.4.2 属性(properties)

算子节点与返回值节点之间关系的属性命名包含两个键值对，其中第一个键名为“return\_order”，值为所连接的返回值节点在当前算子的返回值列表中所处的顺序；第二个键名为“name”，值为所连接的返回值节点的名称。示例如下：{ return \_order: 1, name: “tensor”}。【对于一个返回值没给到name情况下，可命名为null标志】

## 2.5父返回值节点与子返回值节点之间的关系

### 2.4.1 类型(type)

父返回值节点与子返回值节点之间关系的类型命名格式为：“oneOfReturn”。

### 2.4.2 属性(properties)

返回值节点与子返回值节点之间关系的属性命名包含两个键值对，第一个键名为“dtype\_order”，值为所连接的子返回值节点在当前返回值节点的类型列表中所处的顺序；第二个键名为“dtype”，值为所连接的子返回值节点的名称。示例如下：{dtype\_order: 1, dtype: “int”}。

# 算子知识图谱框架间的关系设计

## 3.1 算子节点与算子节点之间的关系

### 3.1.1 类型(type)

一个框架的算子节点与另一个框架相应的算子节点之间关系的类型命名格式为：“equivalentOperator”。

### 3.1.2 属性(properties)

一个框架的算子节点与另一个框架相应的算子节点之间关系的属性包含一个键值对，其中，键名为“framework\_name”，值为所连接的算子节点所属框架的框架名称。示例如下：{framework\_name: “PyTorch”}，{framework\_name: “MindSpore”}，{framework\_name: “PaddlePaddle”}。

## 3.2 父参数节点与父参数节点之间的关系

### 3.2.1 类型(type)

一个框架的算子的参数与另一个框架相应的算子的参数节点之间关系的类型命名格式为：“equivalentParameter”。

### 3.2.2 属性(properties)

一个框架的算子节点参数与另一个框架相应的算子节点参数之间关系的属性包含一个键值对，其中，

键名为“framework\_name\_first”，值为所连接的算子节点1所属框架的框架名称。

键名为“framework\_name\_second”，值为所连接的算子节点2所属框架的框架名称。

键名为“operator\_name\_first”，值为所连接的算子节点1名称。

键名为“operator\_name\_second”，值为所连接的算子节点2名称。

键名为“parameter\_name\_first”，值为所连接的算子节点1的参数名称。

键名为“parameter\_name\_second”，值为所连接的算子节点2的参数名称。

示例如下：{ framework\_name\_first: “PyTorch”，framework\_name\_second: “MindSpore”，operator\_name\_first: “torch.abs”, operator\_name\_second: “mindspore.ops.Abs.abs”, parameter\_name\_first :” input”, parameter\_name\_second :” x”}

## 3.3 父返回值节点与父返回值节点之间的关系

### 3.3.1 类型(type)

一个框架的算子的返回值与另一个框架相应的算子的返回值节点之间关系的类型命名格式为：“equivalentReturn”。

### 3.3.2 属性(properties)

一个框架的算子节点返回值与另一个框架相应的算子节点返回值之间关系的属性包含一个键值对，其中，

键名为“framework\_name\_first”，值为所连接的算子节点1所属框架的框架名称。

键名为“framework\_name\_second”，值为所连接的算子节点2所属框架的框架名称。

键名为”operator\_name\_first”，值为所连接的算子节点1名称。

键名为”operator\_name\_second”，值为所连接的算子节点2名称。

键名为“return\_order\_first”，值为所连接的算子节点1的返回值所属顺序。

键名为“return\_order\_second”，值为所连接的算子节点2的返回值所属顺序。

示例如下：{ framework\_name\_first: “PyTorch”，framework\_name\_second: “MindSpore”， operator\_name\_first: “torch.abs”, operator\_name\_second: “mindspore.ops.Abs.abs”, return\_order\_\_first :1, return\_order\_second :1}