**算子知识图谱设计文档v1.3**

1. 算子知识图谱节点设计
   1. 框架节点
      1. 标签(label)

框架节点的标签命名格式为：“framework”。

* + 1. 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‘，值为所采用的框架；第二个键名为’version’，值为该框架的版本。示例如下：{name: ‘pytoch’, version: ‘1.5.0’}, {name: ‘mindspore’, version: ‘1.5.0’}, {name: ‘paddlepaddle’, version: ‘1.5.0’}。

* 1. 类节点
     1. 标签(label)

框架节点的标签命名格式为：“框架名\_class”，框架名是所采用的代码框架。示例如下：torch\_class, ms\_class, pp\_class。

* + 1. 属性(properties)

框架节点的属性命名包含一个键值对，其中，第一个键名为’name’，值为相应的类名称。示例如下：{name: ‘torch’}, {name: ‘nn’}。

* 1. 算子节点
     1. 标签(label)

算子节点的标签命名格式为：“框架名\_operator”，框架名是所采用的代码框架。示例如下：torch\_operator, ms\_operator, pp\_operator。

* + 1. 属性(properties)

算子节点的属性命名包含一个键值对，第一个键值对的键名为’name’，值为相应的算子名称(此处不用存完整的算子名，完成的算子名称通过类与算子之间的层次关系获取)。示例如下：{name: ‘Conv2d’ }。

* 1. 参数节点
     1. 标签(label)

参数节点根据是否有多个可选类型可以分为父参数节点和子参数节点，其中，父参数节点的标签命名格式为：“框架名\_parameter”，子参数节点的标签命名格式为“框架名\_parameter\_child”，框架名是所采用的代码框架。示例如下：torch\_parameter, ms\_ parameter, pp\_ parameter, torch\_parameter\_child, ms\_ parameter\_child, pp\_ parameter\_child。

* + 1. 属性(properties)

父参数节点的属性命名包含四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。

第一个必需的键值对的键名为’ operator，值为该参数所处的operator名称；第二个必需键值对的键名为’order’，值为该参数在所处算子中的顺序序号；第三个必需键值对的键名为’name’，值为相应的参数名称；第四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：{ operator: 'Conv2d', order: 1, name: 'in\_channels', dtype: 'int'}。

若该参数有多个可选类型，则将该参数节点细分成多个不同的子参数节点。其中父参数节点的’dtype’属性的值重定义为参数类型的数目，子参数节点再添加一维属性，键名为’type’，值为该类型在类型列表中所处的序号。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{ operator: 'Conv2d', order: 3, name: 'kernel\_size', dtype: 2}，两个子参数节点的属性分别为{ operator: 'Conv2d', type: 1, order: 3, name: 'kernel\_size', dtype : 'int'}, { operator: 'Conv2d', type: 2, 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；第二个非必需键值对的键名为’optional’，值名为‘True ‘，表明该参数在相应的算子中是一个可选项。示例如下：父参数节点的属性为{ operator: 'Conv2d', order: 4, name: 'stride', dtype: 2, default: 1, optional: True}，两个子参数节点的属性分别为{ operator: 'Conv2d', type: 1, order: 4, name: 'stride', dtype: 'int', default: 1, optional: True}，{ operator: 'Conv2d', type: 2, order: 4, name: 'stride', dtype: 'tuple', optional: True}。

1. 算子知识图谱框架内关系设计
   1. 框架节点与类节点之间的关系
      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”。

* + 1. 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’class\_name’，值为所连接的算子类节点的名称。示例如下：{class\_name: ‘torch’}。

* 1. 类节点与类节点之间的关系
     1. 类型(type)

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

* + 1. 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’class\_name’，值为所连接的算子类节点的名称。示例如下：{class\_name: ‘nn’}。

* 1. 类节点与算子节点之间的关系
     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”。

* + 1. 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’operator\_name’，值为所连接的算子节点的名称。示例如下：{ operator \_name: ‘Conv2d’}。

* 1. 算子节点与参数节点之间的关系
     1. 类型(type)

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

* + 1. 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’parameter\_name’，值为所连接的参数节点的名称。示例如下：{ parameter\_order: 1, parameter\_name: ‘in\_channels’}。

* 1. 参数节点与子参数节点之间的关系
     1. 类型(type)

参数节点与子参数节点之间关系的类型命名格式为：“oneOf”。

* + 1. 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’ parameter\_type’，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’ parameter\_dtype’，值为所连接的子参数节点的名称。示例如下：{ parameter\_type: 1, parameter\_dtype: ‘int’ }。

1. 算子知识图谱框架间关系设计
   1. 算子节点与算子节点之间的关系
      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：“equivalentClass”。

* + 1. 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命名暂无。

* 1. 参数节点与参数节点之间的关系（暂时不用考虑）
     1. 类型(type)

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

* + 1. 属性(properties)

一个框架的参数节点与另一个框架相应的参数节点之间关系的属性命名暂无。