

GUNDAM::VF2

概述

GUNDAM::VF2是基于自定义点相等和边相等的子图同构匹配算法，除了多了candidate size order之外，其他在算法层面上与传统VF2基本相同

与传统VF2的区别

- 1.可支持自定义点相等以及边相等
- 2.可匹配带有多重边的pattern
- 3.基于VF2的match order策略（选择匹配点的邻接点）的基础上，添加了candidate size order

使用要求

要求pattern具有的接口

```
VertexBegin(),VertexCBegin()  
OutEdgeBegin(),OutEdgeBegin(label),OutEdgeCBegin(),OutEdgeCBegin(label)  
InEdgeBegin(),InEdgeBegin(label),InEdgeCBegin(),InEdgeCBegin(label)  
CountInEdge(),CountOutEdge()
```

要求data graph具有的接口

```
VertexBegin(),VertexCBegin()  
OutEdgeBegin(),OutEdgeBegin(label),OutEdgeCBegin(),OutEdgeCBegin(label)  
InEdgeBegin(),InEdgeBegin(label),InEdgeCBegin(),InEdgeCBegin(label)  
CountInEdge(),CountOutEdge()
```

可用功能

VF2(const QueryGraph &query_graph, const TargetGraph &target_graph,VertexCompare vertex_comp, EdgeCompare edge_comp, MatchCallback user_callback)

自定义点相等和边相等的VF2

VF2(const QueryGraph &query_graph, const TargetGraph &target_graph,MatchCallback user_callback)

基于label相等语义下的vf2

VF2(const QueryGraph &query_graph, const TargetGraph &target_graph,VertexCompare vertex_comp, EdgeCompare edge_comp,int max_result)

自定义点相等和边相等的语义，且计算匹配数是否超过max_result的vf2

VF2(const QueryGraph &query_graph, const TargetGraph &target_graph,int max_result)

基于label相等语义下,且计算匹配数是否超过max_result的vf2

VF2(const QueryGraph &query_graph, const TargetGraph &target_graph,VertexCompare vertex_comp, EdgeCompare edge_comp, int max_result,ResultContainer &match_result)

自定义点相等和边相等,保留至多max_result个匹配结果的VF2，结果存储在match_result里

VF2(const QueryGraph &query_graph, const TargetGraph &target_graph,int max_result, ResultContainer &match_result)

基于label相等语义下,保留至多max_result个匹配结果的VF2，结果存储在match_result里

VF2(const QueryGraph &query_graph, const TargetGraph &target_graph,const typename QueryGraph::VertexType::IDType query_id,const typename TargetGraph::VertexType::IDType target_id,VertexCompare vertex_comp, EdgeCompare edge_comp,int max_result,ResultContainer &match_result)

自定义点相等和边相等,将(query_id,target_id)匹配之后，保留至多max_result个匹配结果的VF2，结果存储在match_result里

未实现部分

- 1.未实现单机多线程版本
- 2.不能支持基于部分匹配计算match
- 3.不支持计算pattern中部分点的match
- 4.不支持用户自定义match order
- 5.不支持用户指定优先匹配哪些节点
- 6.可利用user_callback计算supp,但是效率较低

GUNDAM::VF2_Label_Equal

概述

GUNDAM::VF2_Label_Equal是基于label相等的子图同构匹配算法，与GUNDAM::VF2相比，增加了很多优化方法，且可实现更多功能，同时也需要data_graph添加一些索引

与传统VF2的区别

- 1.可匹配带有多重边的pattern
- 2.基于VF2的match order策略（选择匹配点的邻接点）的基础上，添加了candidate size order
- 3.采用了DAF中的DAGDP来减少candidate_set大小
- 4.采用了动态更新candidate_set的策略

与GUNDAM::VF2的区别

- 1.加强了InitCandidateSet
- 2.增加了RefineCandidateSet
- 3.增加了UpdateCandidateSet
- 4.增加了prune_callback和update_callback，可以实现更多功能

与其他子图同构算法的区别

DAF

与DA-cand算法除了不用拓扑序而是用邻接点之外，其他基本相同

使用要求

要求pattern具有的接口

VertexBegin(),VertexCBegin()
OutEdgeBegin(),OutEdgeBegin(label),OutEdgeCBegin(),OutEdgeCBegin(label)
InEdgeBegin(),InEdgeBegin(label),InEdgeCBegin(),InEdgeCBegin(label)
CountInEdge(),CountInEdge(label),CountOutEdge(),CountOutEdge(label)
CountInVertex(),CountInVertex(label),CountOutVertex(),CountOutVertex(label)

要求data graph具有的接口

VertexBegin(),VertexCBegin()
OutEdgeBegin(),OutEdgeBegin(label),OutEdgeBegin(label,ptr),OutEdgeCBegin(),OutEdgeCBegin(label),OutEdgeCBegin(label,ptr)
InEdgeBegin(),InEdgeBegin(label),InEdgeCBegin(label,ptr),InEdgeCBegin(),InEdgeCBegin(label),InEdgeCBegin(label,ptr)
OutVertexBegin(),OutVertexBegin(label),OutVertexCBegin(),OutVertexCBegin(label)
InVertexBegin(),InVertexBegin(label),InVertexCBegin(),InVertexCBegin(label)
CountInEdge(),CountInEdge(label),CountOutEdge(),CountOutEdge(label)
CountInVertex(),CountInVertex(label),CountOutVertex(),CountOutVertex(label)

可用功能

VF2_Label_Equal(const QueryGraph &query_graph,const TargetGraph &target_graph,MatchCallback user_callback)

自定义user_callback的vf2

VF2_Label_Equal(const QueryGraph &query_graph,const TargetGraph &target_graph, int max_result)

计算匹配数是否超过max_result的vf2

VF2_Label_Equal(const QueryGraph &query_graph,const TargetGraph &target_graph, int max_result,ResultContainer &match_result)

保留至多max_result个匹配结果的VF2， 结果存储在match_result里

VF2_Label_Equal(const QueryGraph &query_graph,const TargetGraph &target_graph,MatchCallback user_callback,PruneCallback prune_callback,UpdateCandidateCallback update_candidate_callback)

自定义匹配存储处理 (user_callback)， 中间过程的人为剪枝处理(prune_callback),candidate_set的人为处理 (update_candidate_callback)的vf2

VF2_Label_Equal(const QueryGraph &query_graph, const TargetGraph &target_graph,typename QueryGraph::VertexConstPtr cal_supp_vertex_ptr,const std::vector &possible_supp,std::vector &supp)

计算cal_supp_vertex_ptr的supp的vf2， 其中possible_supp表示有可能成为supp的候选结果

VF2_Label_Equal(const QueryGraph &query_graph, const TargetGraph &target_graph,const typename QueryGraph::VertexType::IDType query_id,const typename TargetGraph::VertexType::IDType target_id, int max_result,ResultContainer &match_result)

将(query_id,target_id)匹配之后， 保留至多max_result个匹配结果的VF2， 结果存储在match_result里

未实现部分

- 1.未实现单机多线程版本
- 2.未有直接的基于部分匹配计算match接口
- 3.不支持用户自定义match order
- 4.不支持用户指定优先匹配哪些节点
- 5.未有通用且高效的直接计算pivot的supp接口
- 6.目前存在部分pattern匹配过慢的情况,需要修改部分代码写法