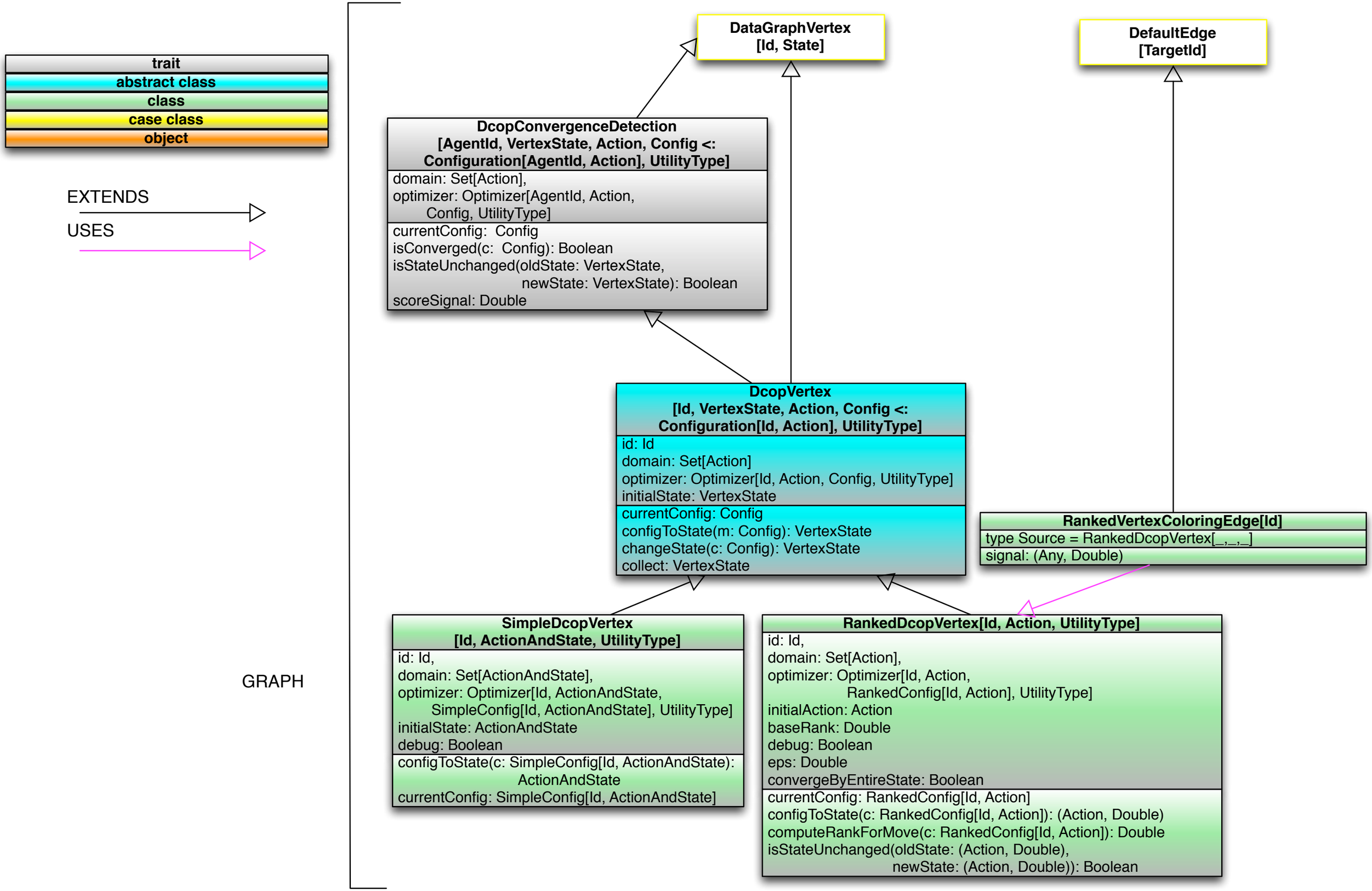
trait
abstract class
class
case class
object
EXTENDS
USES
GRAPH
DataGraphVertex
[Id, State]
DefaultEdge
[TargetId]
DcopConvergenceDetection
[AgentId, VertexState, Action, Config <:
Configuration[AgentId, Action], UtilityType]
domain: Set[Action],
optimizer: Optimizer[AgentId, Action,
Config, UtilityType]
currentConfig: Config
isConverged(c: Config): Boolean
isStateUnchanged(oldState: VertexState,
newState: VertexState): Boolean
scoreSignal: Double
DcopVertex
[Id, VertexState, Action, Config <:
Configuration[Id, Action], UtilityType]
id: Id
domain: Set[Action]
optimizer: Optimizer[Id, Action, Config, UtilityType]
initialState: VertexState
currentConfig: Config
configToState(m: Config): VertexState
changeState(c: Config): VertexState
collect: VertexState
RankedVertexColoringEdge[Id]
type Source = RankedDcopVertex[_,_,_]
signal: (Any, Double)
SimpleDcopVertex
[Id, ActionAndState, UtilityType]
id: Id,
domain: Set[ActionAndState],
optimizer: Optimizer[Id, ActionAndState,
SimpleConfig[Id, ActionAndState], UtilityType]
initialState: ActionAndState
debug: Boolean
configToState(c: SimpleConfig[Id, ActionAndState):
ActionAndState
currentConfig: SimpleConfig[Id, ActionAndState]
RankedDcopVertex[Id, Action, UtilityType]
id: Id,
domain: Set[Action],
optimizer: Optimizer[Id, Action,
RankedConfig[Id, Action], UtilityType]
initialAction: Action
baseRank: Double
debug: Boolean
eps: Double
convergeByEntireState: Boolean
currentConfig: RankedConfig[Id, Action]
configToState(c: RankedConfig[Id, Action]): (Action, Double)
computeRankForMove(c: RankedConfig[Id, Action]): Double
isStateUnchanged(oldState: (Action, Double),
newState: (Action, Double)): Boolean