Определения

$v1,\ v2$ – вершины, между которыми нужно найти путь
$[V, E]$ – ориентированный граф
$V = \{v\}$ – множество вершин
$E = \{(v, w) :\ v, w \in V\}$ – множество рёбер
$v \to w$ означает, что существует путь из v в w

$F_1 = \{v \in V : v1 \to v\}$ – множество вершин, достижимых из $v1$
$T_1 = \{v \in V : v \to v1\}$ – множество вершин, из которых достижима $v1$
$F_2 = \{v \in V : v2 \to v\}$ – множество вершин, достижимых из $v2$
$T_2 = \{v \in V : v \to v2\}$ – множество вершин, из которых достижима $v2$

$\forall v \in V$:
$IN_v = \{w \in V : (w, v) \in E\}$ – множество соседей v по входящим дугам
$OUT_v = \{w \in V : (v, w) \in E\}$ – множество соседей v по исходящим дугам

Правила целостности

$v \in F_1 \Rightarrow OUT_v \subset F_1$ – если v достижима из $v1$, то её соседи по исходящим дугам тоже
$v \in F_2 \Rightarrow OUT_v \subset F_2$ – если v достижима из $v2$, то её соседи по исходящим дугам тоже
$v \in T_1 \Rightarrow IN_v \subset T_1$ – если $v1$ достижима из v, то из её соседей по входящим дугам тоже
$v \in T_2 \Rightarrow IN_v \subset T_2$ – если $v2$ достижима из v, то из её соседей по входящим дугам тоже

Условия существования пути

$v1 \to v2 \Leftrightarrow F_1 \cap T_2 \neq \varnothing$ – условие существования пути из $v1$ в $v2$
$v2 \to v1 \Leftrightarrow F_2 \cap T_1 \neq \varnothing$ – условие существования пути из $v2$ в $v1$

Алгоритм поиска основан на том, что по очереди загружаются новые вершины из двух списков загрузки (один формируется от $v1$, другой от $v2$). Новая вершина добавляется в граф, а множества F_1, F_2, T_1, T_2 приводятся в соответствие правилам целостности. Затем проверяются условия существования пути. Если какое-то из условий выполнено, то путь существует, а все его точки принадлежат соответствующему множеству $F_i \cap T_j$, в котором он ищется обычным способом (как правило, размер этого множества мал). Когда найдены пути в обе стороны, алгоритм перестаёт загружать новые вершины и прекращает работу.