Образец

**ТИТУЛЬНЫЙ ЛИСТ**

**ПО ШАБЛОНУ**

Оглавление

Введение 3

1. Анализ предметной области 3

2. Формализованное описание процесса управления 17

3. Разработка миварной базы знаний (МБЗ) 19

4. Создание миварной экспертной системы (МЭС) 22

5. Проведение тестирования МЭС 41

Заключение 46

Список литературы 46

# **Введение**

Миварные технологии [1] логического искусственного интеллекта [2] активно развиваются с 2002 года, когда был предложен линейной вычислительной сложности алгоритм логического вывода [3]. Для научной области «Умные производственные системы» важно отметить, что миварные технологии применяют для моделирования сложных систем управления технологическими процессами [4] с возможностью распараллеливания алгоритмов в научно-технических и вычислительных задачах [5] и сравнения многомерных векторов [6] для практического применения в АСУ и экспертных системах реального времени [7]. На данный момент важно отметить, что задачи распознавания изображений и понимания образов [8], а также семантического обнаружения объектов [9] успешно решаются на основе миварных экспертных систем для планирования траектории робота и создания систем принятия решений "РОБО!РАЗУМ" групп автономных роботов, комбайнов, тракторов и т.п. [10].

В данной работе мы исследуем границы применимости миварных технологий в задачах планирования маршрута робота в двумерном лабиринте.

Цель работы:

Создание информационной системы с веб-интерфейсом для планирования маршрута робота с применением миварных технологий и классических алгоритмов поиска пути.

Задачи:

* проанализировать распространенные методы планирования маршрутов
* создать и протестировать МБЗ для планирования маршрута робота
* разработать информационную систему с веб-интерфейсом, которая позволит моделировать передвижение автономного робота в лабиринте
* провести эксперимент со сравнением планирования маршрута с применением миварного подхода и других методов

Актуальность:

 Традиционные алгоритмы доказали свою эффективность, однако их применение может быть ограничено в условиях динамически изменяющейся среды, высокой вычислительной сложности или появлении дополнительных условий. В связи с этим актуальным становится поиск альтернативных подходов, способных повысить гибкость планирования траектории.

# **1. Анализ предметной области**

1.1 Автономные роботы и области их применения

Автономные роботы — это механические устройства, способные выполнять задачи без непосредственного вмешательства человека, основываясь на программных алгоритмах, сенсорах и системах искусственного интеллекта. Их способность работать в условиях минимального человеческого контроля делает их незаменимыми в различных отраслях: от промышленного производства и логистики до медицины и сельского хозяйства.

Технологии планирования маршрутов автономных роботов активно изучаются и используются. Рассмотрим несколько таких систем.

Одной из наиболее выдающихся систем является разработка Amazon Kiva, которая представляет собой систему автоматизированных складских роботов, разработанную для оптимизации логистических операций на складах компании Amazon. Основная задача системы — быстрое, точное и эффективное перемещение товаров внутри склада, минимизация затрат времени и ресурсов на поиск и доставку необходимых объектов.



Рисунок 1 – Автономный робот Amazon Kiva

Для планирования маршрутов используются централизованные алгоритмы, которые рассчитывают оптимальный маршрут для каждого робота с учетом загруженности склада. Для работы этой системы предполагается строго структурированное пространство с заранее заданными маршрутами, отмеченными QR-кодами на полу.

Аналогичные системы можно встретить также у отечественных поставщиков. В настоящее время автономные мобильные роботы активно разрабатываются, совершенствуются и внедряются в повседневную жизнь для решения бытовых и промышленных задач.

Кроме доставки и перевозки грузов автономные мобильные роботы применяются также для задач, несущих риск для жизни и здоровья человека. Например, Автономный мобильный робот S6.3 LR предназначен для безопасного обследования нефтегазового оборудования с использованием лазерного газового детектора.



Рисунок 2 – Автономный робот S6.3 LR

Таким образом автономные мобильные роботы нашли себе применение в самых разных сферах жизни человека. В настоящее время разработка мобильных роботов является быстрорастущей и перспективной областью.

1.2. Классификация автономных роботов

Автономные мобильные роботы сегодня представляют собой разнообразный класс технических систем, которые можно классифицировать по множеству взаимосвязанных параметров.

По способу передвижения автономные роботы варьируются от простейших и наиболее распространенных колесных платформ до бионических шагающих конструкций. Кроме того, для мобильных автономных роботов применяются также гусеничные системы, которые открыли новую эру в освоении труднодоступных территорий.

Особый интерес представляют шагающие роботы, которые, имитируя принципы передвижения живых организмов, способны преодолевать такие препятствия, которые остаются непреодолимыми для других типов мобильных платформ. Биоморфные конструкции вроде четвероногих роботов Boston Dynamics или человекоподобных машин Atlas демонстрируют невероятные возможности в плане балансировки и адаптации к сложному рельефу.



Рисунок 3 – Механическая собака от BostonDynamics



Рисунок 4 – Человекоподобный робот Atlas от BostonDynamics

Другим специфичным классом автономных роботов являются летающие и плавающие роботы. Летающие автономные системы варьируются от компактных дронов, способных проникать в самые узкие пространства, до крупных беспилотников. Подводные роботы могут работать в условиях полного отсутствия GPS-навигации и ограниченной связи, поэтому представляют собой отдельный технологический вызов, решаемый с помощью сложных систем навигации. В настоящее время такие роботы применяются главным образом в исследовательских целях.

Роботы также отличаются степенью автономности, наиболее развитыми по этому критерию можно считать полностью автономные системы, принимающие решения на основе сложных алгоритмов искусственного интеллекта без какого-либо вмешательства человека. Полуавтономные системы занимают промежуточное положение, сохраняя возможность вмешательства оператора в критических ситуациях, что особенно важно в некоторых сферах.

1.3. Методы декомпозиции пространства

Планирование маршрута робота — сложная задача, требующая комбинации алгоритмов и технологий для обеспечения безопасности и оптимальности пути. Основные подходы делятся на **глобальное** и **локальное** планирование, дополняемые методами навигации и обработки данных в реальном времени.

Глобальное планирование выполняется на основе заранее известной карты среды со статическими препятствиями. Локальное планирование корректирует маршрут в реальном времени, используя данные с датчиков.

В обоих случаях требуется декомпозиция пространства. Так как эта работа предполагает глобальное планирование маршрута, рассмотрим наиболее популярные способы декомпозиции пространства, которые могут быть применены для решения нашей задачи.

1.3.1. Построение путей на основе диаграммы Вороного

Использование диаграмм вороного требует знания расположения всех препятствий в пространстве, поэтому он относится к глобальным методам планирования.

По определению диаграмма Вороного является разбиением плоскости, выполненным таким образом, что для каждого элемента конечного множества точек N справедливо следующее утверждение: расстояние от каждой точки из множества M, образованного разбиением вокруг точки N[i] меньше, чем расстояние до любой другой точки из N. Каждая ячейка такого разбиения называется локусом.

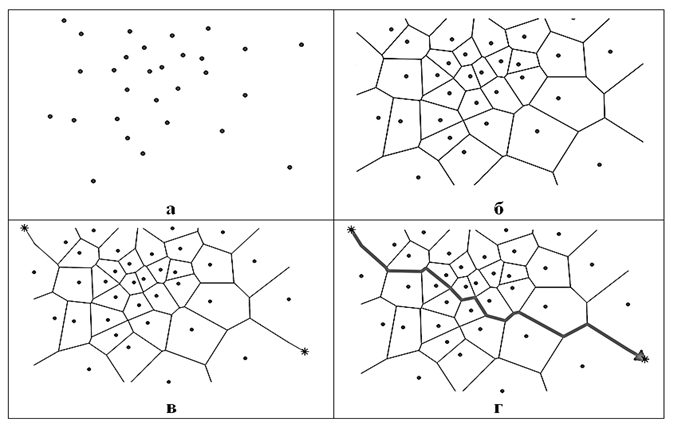


Рисунок 5 – Построение маршрута с использованием диаграммы Вороного

В случае построения маршрута по диаграмме Вороного, точки множества N – это точки препятствия, а точки M (стороны многоугольников) будут ближайшими относительно препятствий.

Диаграммы Вороного широко применяются для построения безопасных путей и оптимизации расположения объектов, например, вышек сотовой связи. Однако он плохо применим для построения динамических маршрутов.

1.3.2. Клеточная декомпозиция

Клеточная декомпозиция является еще одним примером метода для решения задач глобального планирования маршрута.

Суть клеточной декомпозиции заключается в том, что пространство разбивается на дискретные ячейки (клетки), после чего поиск пути сводится к навигации по графу, где вершины — это клетки, а рёбра — возможные переходы между ними. Это преобразование пространства в регулярную структуру открывает возможность применения эффективных алгоритмов поиска, таких как A\* или волновой алгоритм, которые работают именно с дискретными представлениями данных.

Основное преимущество клеточной декомпозиции заключается в ее универсальности и относительной простоте реализации. В зависимости от конкретной задачи можно выбрать разные подходы к разбиению пространства. Например, регулярная декомпозиция с одинаковыми квадратными ячейками отлично подходит для простых случаев, когда важна скорость работы и простота реализации. Однако в более сложных сценариях, где требуется высокая точность при наличии множества препятствий сложной формы, лучше себя показывает адаптивная декомпозиция, которая динамически изменяет размер ячеек в зависимости от сложности участка пространства.

1.4. Виды лабиринтов

Слово "лабиринт" в первоначальном смысле означает лишь одну извилистую дорожку, которая не имеет ответвлений и не пересекается с другими ходами. Однако в рамках этой работы мы будем считать лабиринтом произвольную пространственную структуру, содержащую конечное количество путей между разными точками пространства.

1.4.1. Размерность лабиринтов

Лабиринты можно классифицировать по их пространственной организации.

* Двухмерные лабиринты представляют собой структуры, отображаемые на плоскости. Они состоят из сетки ячеек, где часть из них является стенами, а остальные — проходами. Такие лабиринты широко применяются в игровом дизайне и головоломках благодаря простоте визуализации и навигации.
* Трёхмерные лабиринты включают несколько уровней, что добавляет дополнительное измерение — перемещение вверх и вниз. Это усложняет структуру и требует более сложных алгоритмов генерации.

1.4.2. Топология лабиринтов

Топология определяет геометрические свойства пространства лабиринта.

* Обычная (эвклидова) топология предполагает стандартное представление лабиринта на плоскости или в трёхмерном пространстве без искажений.
* Необычная топология включает лабиринты, существующие на нестандартных поверхностях, таких как куб, тор или лента Мёбиуса. Такие структуры могут содержать петли, замкнутые пути и другие топологические аномалии, усложняющие навигацию.

1.4.3. Тесселяция (разбиение пространства)

Тесселяция определяет форму ячеек, из которых состоит лабиринт.

* Ортогональная тесселяция использует прямоугольные ячейки, расположенные в сетке. Это наиболее распространённый тип благодаря простоте генерации.
* Дельта-тесселяция основана на треугольных ячейках, что создаёт менее предсказуемую структуру.
* Сигма-тесселяция применяет шестиугольные ячейки, обладающие уникальными свойствами, такими как отсутствие явных углов.
* Crack-тесселяция представляет собой аморфную структуру с ячейками произвольной формы, что делает лабиринт хаотичным.
* Фрактальная тесселяция предполагает рекурсивное вложение лабиринтов меньшего масштаба, создавая самоподобные структуры.

1.4.4. Маршрутизация и структура проходов

Наиболее важным для нашей работы критерием классификации является организация путей внутри лабиринта.

* Идеальные лабиринты не содержат петель и изолированных областей, гарантируя единственный путь между любыми двумя точками. Они часто генерируются алгоритмами Прима или Крускала.
* Плетёные лабиринты допускают петли и множественные маршруты, что увеличивает сложность прохождения.
* Унимодальные лабиринты имеют лишь один извилистый путь без развилок, что делает их полезными для медитативных и декоративных целей.
* Частично решаемые лабиринты содержат ложные пути и несколько выходов, из которых только один является верным.
* Динамические лабиринты меняют свою структуру во времени или в зависимости от действий пользователя, например, перемещающиеся стены или открывающиеся двери.

В нашей работе мы будем рассматривать двумерные лабиринты с ортогональной тасселяцией. По структуре рассмотрим несколько видов лабиринта: плетеный лабиринт и идеальный.

1.5. Алгоритмы поиска пути

Поиск пути в лабиринте представляет собой фундаментальную задачу в области теории графов, искусственного интеллекта и робототехники. Различные алгоритмы предлагают разнообразные подходы к решению этой задачи, отличающиеся по вычислительной сложности, требованиям к памяти и условиям применимости.

1.5.1. Поиск в глубину (DFS)

Поиск в глубину основан на рекурсивном или итеративном исследовании одной ветви пути до её завершения с последующим возвратом (бэктрекингом). Этот метод использует стековую структуру, что делает его реализацию относительно простой. Однако DFS не гарантирует нахождения кратчайшего пути, поскольку первым обнаруживает любой достижимый маршрут. В худшем случае временная сложность составляет O(V + E), где V — количество вершин, а E — рёбер. Основным недостатком является потенциальная неэффективность в больших лабиринтах с циклами, если не применяется проверка посещённых узлов.

1.5.2. Поиск в ширину (BFS)

В отличие от DFS, поиск в ширину исследует все соседние узлы перед углублением в следующие уровни, используя очередь. Это гарантирует нахождение кратчайшего пути в невзвешенном графе. Алгоритм требует больше памяти, поскольку хранит все узлы текущего уровня, но его временная сложность также оценивается как O(V + E). BFS широко применяется в задачах, где критична минимальная длина пути.

1.5.3. Алгоритм Дейкстры

Алгоритм Дейкстры расширяет BFS, учитывая веса рёбер. Он использует очередь с приоритетом для выбора узла с минимальной текущей стоимостью пути. Временная сложность составляет O((V + E) log V), что делает его менее эффективным по сравнению с BFS в отсутствие весов. Однако в лабиринтах с переменной стоимостью перемещения (например, разными типами местности) Дейкстра остаётся оптимальным выбором.

1.5.4. Алгоритм A\*

A\* комбинирует подход Дейкстры с эвристической функцией h(n), оценивающей оставшееся расстояние до цели. Функция стоимости f(n) = g(n) + h(n), где g(n) — пройденный путь, позволяет алгоритму эффективно направлять поиск. Если эвристика допустима (не переоценивает реальное расстояние), A\* находит кратчайший путь. Время работы зависит от качества эвристики, в лучших случаях значительно превосходя Dijkstra.

1.5.6. Алгоритм Ли (волновой алгоритм)

Волновой алгоритм, по сути, является вариацией BFS, распространяющей "волну" значений расстояний от стартовой точки. Он гарантирует кратчайший путь, но требует O(N²) памяти для хранения матрицы расстояний, что ограничивает его применение в больших лабиринтах.

1.5.7. Theta\*

Theta\* — это оптимизация алгоритма A, предназначенная для поиска пути в непрерывных или градиентных пространствах (например, в двумерных или трёхмерных средах без строгой сеточной структуры). В отличие от классического A, который строит путь строго по узлам сетки, Theta\* позволяет траектории свободно проходить между вершинами, если между ними существует прямая видимость.

При переходе от одного узла к другому Theta\* проверяет, можно ли соединить текущую вершину с какой-либо из предыдущих напрямую, без промежуточных узлов. Если такая прямая траектория существует, алгоритм "перепрыгивает" через ненужные узлы, сокращая путь. В отличие от A, где каждый следующий узел обязательно связан с предыдущим, Theta может изменять родителя узла, если найден более оптимальный вариант через другую вершину. Путь, найденный Theta\*, не только кратчайший по расстоянию, но и более гладкий (имеет меньшее количество переходов), что особенно важно в робототехнике и компьютерных игр

1.5.8. Lazy Theta\*

Lazy Theta\* — это оптимизированная версия Theta. В отличие от Theta, Lazy Theta откладывает проверку прямой видимости между узлами до момента их извлечения из очереди, что снижает вычислительные затраты.

Вместо проверки на каждом шаге алгоритм определяет, можно ли соединить текущий узел с родителем прямой линией, только при его обработке. Он особенно полезен в robotics и 3D-навигации, где важна скорость без значительной потери точности.

1.5.9. Миварная логика

Миварный подход уникален тем, что позволяет обеспечить линейную вычислительную сложность логического вывода за счет того, что база данных, логический вывод и вычислительная обработка представляют собой единое целое. Есть примеры успешного применения миварных технологий для решения задач планирования маршрута.

**2. Формализация**

Формализацию предметной области планирования маршрута можно разделить условно на две части: формализация лабиринта и формализация самого процесса планирования маршрута.

Существует множество способов представления лабиринта для программной обработки. Наиболее распространенные – графовый и матричный.

Для нашей программы мы используем простой лабиринт со стенами и проходами. Поэтому в графовом виде он мог быть представлен следующим образом:

* Вершины графа- точки пространства.
* Рёбра - пути между вершинами.

Однако мы выбрали матричный вид представления как наиболее простой для отображения и понятный в обработке. Он представляет собой матрицу, представляющую собой все точки пространства и содержащую 0 и 1, где 0 – это пороход, а 1 – это препятствие.

Для получения миварной базы знаний необходимо формализовать также процесс построения маршрута. В этом смысле есть несколько способов.

2.1. Хранятся только проходимые точки

Суть способа заключается в том, хранятся исключительно проходимые точки и переходы между ними. В этом случае все, что нужно сделать – прописать правила переходов между каждой парой точек:

Пусть есть две точки, между которыми есть прямой путь: A и B, тогда правила перехода между ними будут иметь следующий вид:

* + A->B = path(A) + 1
  + B->A = path(B) + 1,

где path(x) есть количество шагов, выполненное алгоритмов для попадания в точку х.

2.2. Хранится признак проходимости точки

При использовании этого способа хранятся все точки пространства с описанием всех прямых переходов между ними, вне зависимости от того, является ли точка проходимой или нет. Кроме того, хранится признак проходимости каждой точки пространства. Таким образом переходы между любыми двумя точками А и B будут описаны следующим условием:

* Если (А проходима) И (B проходима), то А->B = path(A)+1
* Если (B проходима) И (А проходима), то B->A = path(B)+1,

где path(x) есть количество шагов, выполненное алгоритмов для попадания в точку х.

В нашей работе мы применяем способ, описанный в пункте 2.1, так как при использовании этого метода в МБЗ хранится меньший объем данных.

# **3. Разработка миварной базы знаний (МБЗ)**

# **4. Создание миварной экспертной системы (МЭС)**

# **5. Проведение тестирования МЭС**

Тестирование 1. Подача листовой стали на удаление заусенцев:

Запускаем линию производства автомобильных колесных дисков, после того, была пройдена проверка на отсутствие повреждений лист переноситься на удаление заусенцев, на этапе идет проверка на удаление и в случае успеха лист направляется на маркировку . На Рисунке 9 показано тестирование в КЭСМИ.

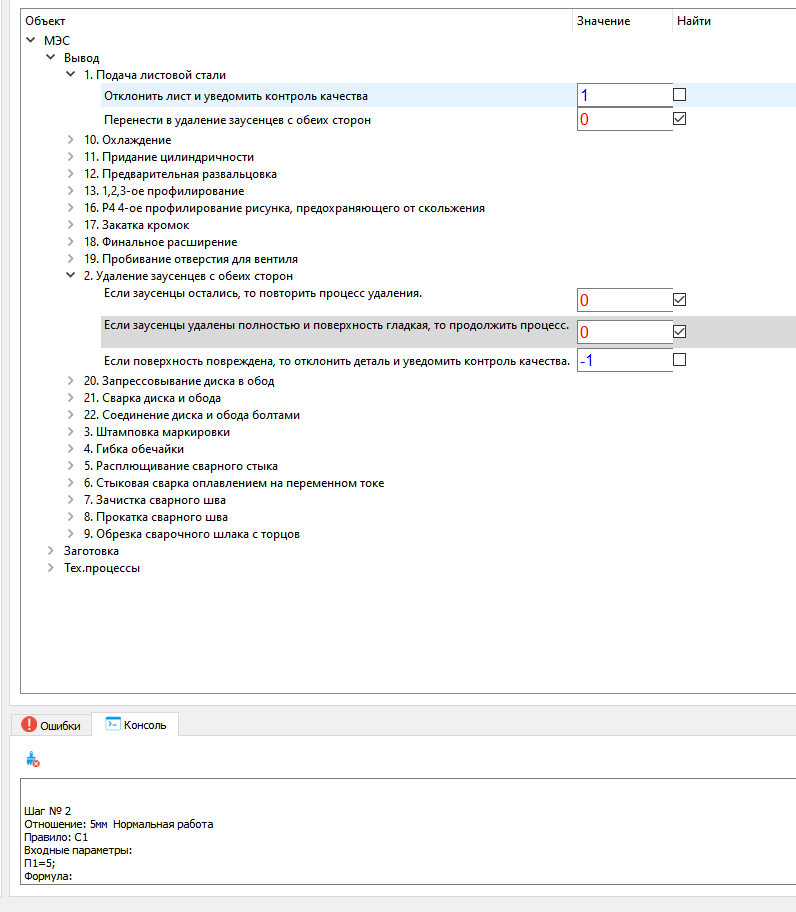


Рисунок 9. Тестирование сварки диска и обода.

Далее, получаем граф тестирования системы сварки диска и обода.

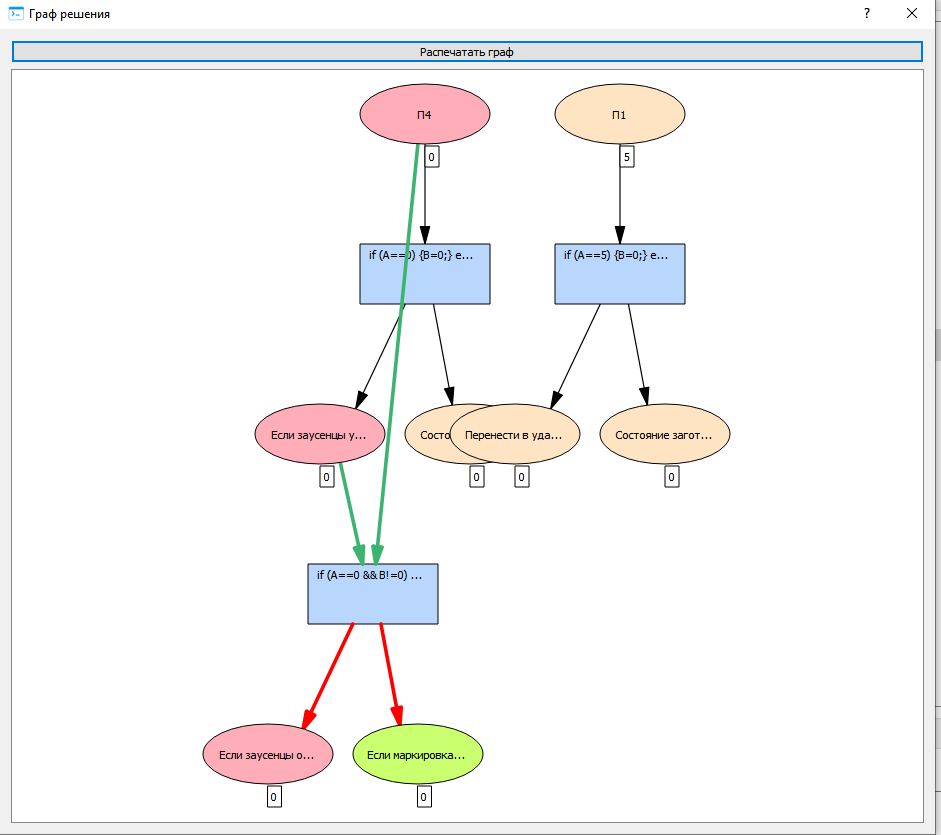


Рисунок 10. Граф тестирования системы сварки диска и обода.

Тестирование 2. Подача листовой стали на удаление заусенцев с аварией:

Запускаем линию производства автомобильных колесных дисков, после того, была пройдена проверка на отсутствие повреждений лист переносится на удаление ,где был выполнен повторная проверка на повреждение после чего было донесено уведомление отделу качества ,а заготовка выводится с производства .На Рисунке 11 показано тестирование в КЭСМИ.

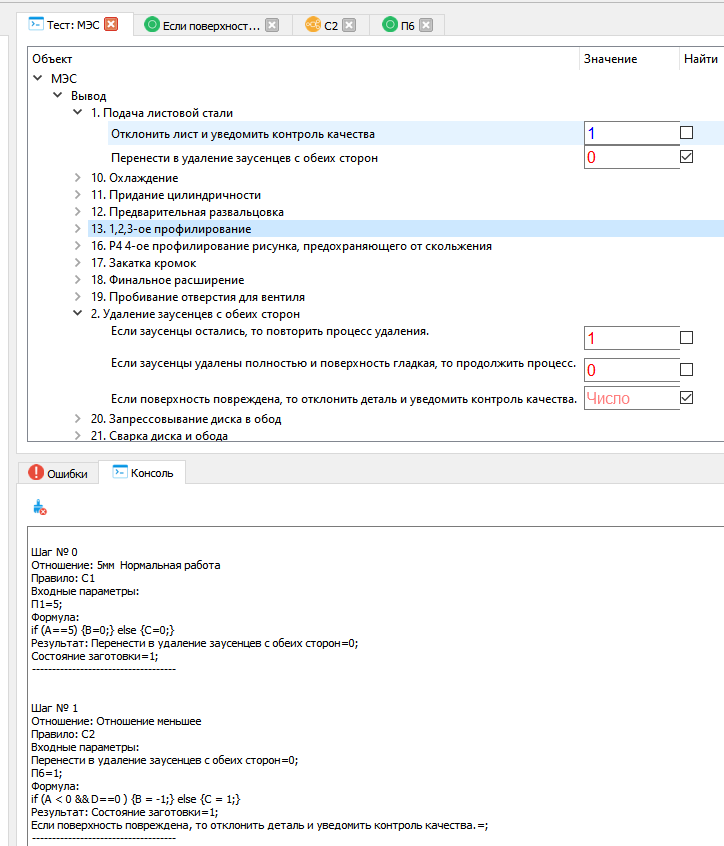


Рисунок 11. Тестирование системы 4-го профилирования рисунка, предохраняющего от скольжения

Далее, получаем граф тестирования системы 4-го профилирования рисунка, предохраняющего от скольжения

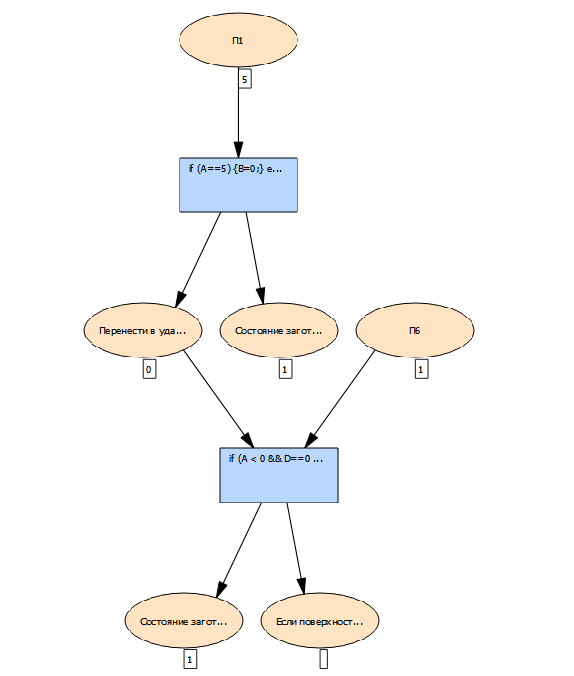


Рисунок 12. Граф тестирования системы 4-го профилирования рисунка, предохраняющего от скольжения

Тестирование 3. Подача листовой стали на удаление заусенцев и маркировку с аварией:

Запускаем линию производства автомобильных колесных дисков, после того, была пройдена проверка на отсутствие повреждений лист переноситься на удаление заусенцев, на этапе идет проверка на удаление и в случае успеха лист направляется на маркировку, при обнаружение неверной маркировки деталь снимают с производства и уведомляют отдел качества. На Рисунке 14 показано тестирование в КЭСМИ.

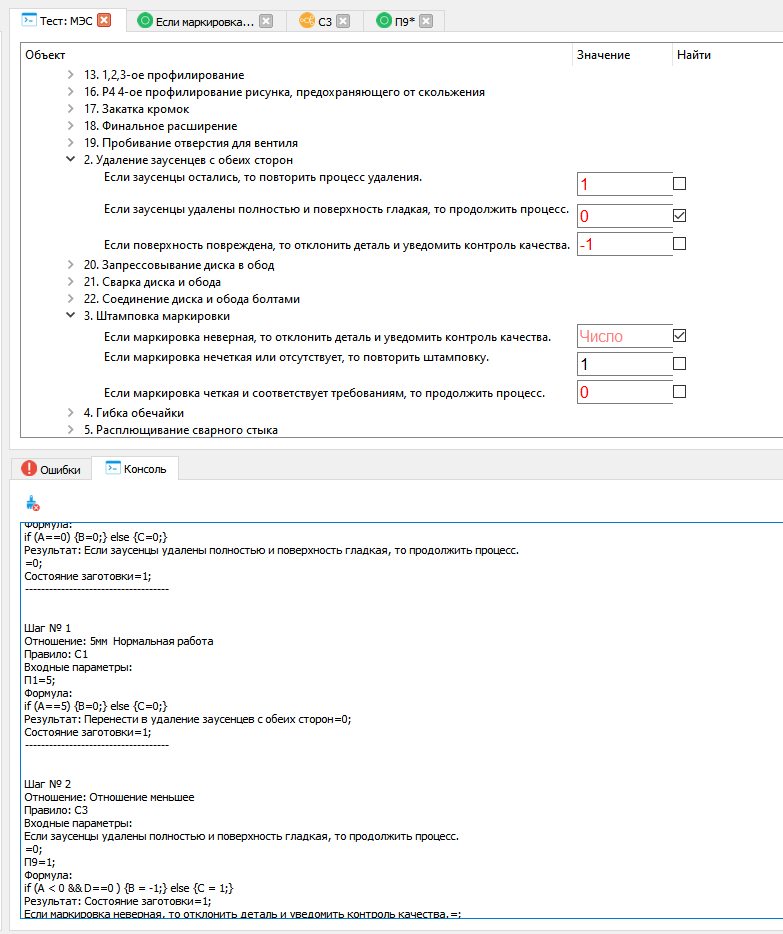


Рисунок 14. Тестирование системы охлаждения.

Далее, получаем граф тестирования системы охлаждения.

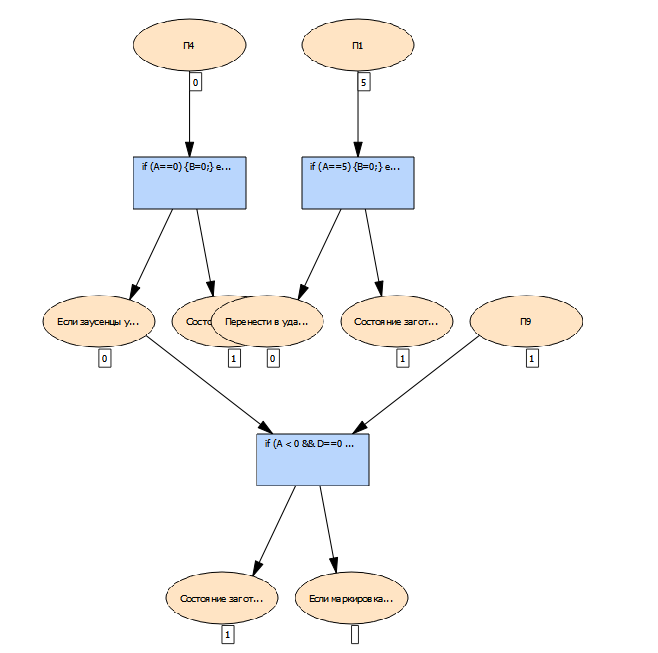


Рисунок 12. Граф тестирования охлаждения

# **Заключение**

Вывод: Мы применили средства автоматизации к станку для стыковой сварки, для улучшения качества производства колесных дисков

Цель достигнута

Задачи

1

2

3

4

5

решены.

Поподробнее

# **Список литературы**

сначала про миварные технологии, потом про специфику предметной области

1. Варламов О.О. Миварные технологии как некоторые направления искусственного интеллекта // Проблемы искусственного интеллекта. 2015. № 1(1). С. 23-37. EDN WDNPGZ.
2. Варламов О.О. Автоматизация умственной деятельности людей через логический искусственный интеллект как фундаментальный механизм развития или гибели человечества // Проблемы искусственного интеллекта. 2017. № 3(6). С. 23-31. EDN YNTRSV.
3. Варламов О.О. Эволюционные базы данных и знаний для адаптивного синтеза интеллектуальных систем. Миварное информационное пространство. М.: «Радио и связь», 2002. 286 с. EDN RWTCOP.
4. Сергушин Г.С. и др. Исследование возможностей информационного моделирования сложных систем управления технологическими процессами на основе миварных технологий // Автоматизация и управление в технических системах. 2013. № 2(4). С. 51-66. EDN RDWXUT.
5. Владимиров А.Н. и др. Применение многопроцессорного вычислительного кластера НИИР для распараллеливания алгоритмов в научно-технических и вычислительных задачах // Труды НИИ Радио. 2009. № 3. С. 120-123. EDN KYNLNN.
6. Семенов А. А. Исследование способов подбора рекламных кампаний на основе сравнения многомерных векторов // Проблемы искусственного интеллекта. 2020. № 1(16). С. 89-104. EDN UEBEPL.
7. Варламов О.О., Санду Р.А., Владимиров А.Н. и др. Миварный метод логико-вычислительной обработки информации для АСУ, тренажеров, экспертных систем реального времени и архитектур, ориентированных на сервисы // Искусственный интеллект. 2010. № 4. С. 558-565. EDN TIFHLT.
8. Мащенко Е.И., Карпов Д.К. и др. Создание миварной экспертной системы для понимания образов и принятия решений при обнаружении падений людей // Проблемы искусственного интеллекта. 2024. № 4(35). С. 88-100. DOI 10.24412/2413-7383-2024-4-88-100. EDN FGLHZP.
9. Шэнь Ц., Гун Ш. и др. Динамическое планирование траектории робота на основе семантического обнаружения объектов с использованием миварной экспертной системы // Проблемы искусственного интеллекта. 2024. № 4(35). С. 164-176. DOI 10.24412/2413-7383-2024-4-164-176. EDN DHVOFC.
10. Варламов О.О. О создании на основе миварных систем принятия решений "РОБО!РАЗУМ" групп автономных комбайнов и тракторов для сельского хозяйства // Проблемы искусственного интеллекта. 2019. № 2(13). С. 49-62. EDN AMUYCK.
11. <https://owen.ru/product/dtp_termopari_na_osnove_ktms_s_kommutatcionnoj_golovkoj/equipment>
12. <https://owen.ru/product/plk210_polygon/connection>
13. <https://www.metsol.ru/info/articles/sistema-slezheniya-za-svarnym-shvom-th6i/?ysclid=m4msjcc38o799766278>
14. <http://wheelmachinery.ru/profile/wheel-manufacturing/199771/0/>

Коценко А.А., Манцев А. А., Денисов А.С., Лосева С.С., Варламов О.О. Применение миварных технологий для управления автономными агентами в однопользовательской компьютерной игре // Естественные и технические науки 2024 .- № 5 (192) .- С. 216 – 224 DOI: [10.25633/ETN.2024.05.18](https://doi.org/10.25633/ETN.2024.05.18)

<http://www.sibgazovik.ru/news/association/2022-09-12/>

<https://www.aboutamazon.com/news/operations/amazon-robotics-robots-fulfillment-center>

<https://vektorus.ru/blog/mobilnye-roboty.html>

<https://bostondynamics.com/>

Попова А. А. Глобальное планирование маршрута мобильного робота на основе графовых методов // Вестник Астраханского государственного технического университета. Серия: Управление, вычислительная техника и информатика. 2024. № 2. С. 57–67. https://doi.org/10.24143/2072-9502-2024-2-57-67. EDN BEIHAL.

https://habr.com/ru/articles/445378/