Разработка верификатора автоматных программ

Егоров К. В., гр. 4538

Руководитель Шалыто А. А.

Цель работы

- Разработка верификатора моделей автоматных программ, созданных при помощи инструментального средства *UniMod*.
  - Требования записываются на языке логики линейного времени (*Linear Time Logic, LTL*).
  - Оптимизация работы верификатора на многоядерных компьютерах.

Понятие верификации

Верификация – метод проверки того, что программа соответствует заданной спецификации (обладает необходимыми свойствами).

Область применения

- Системы управления транспортом.
- Медицинское оборудование.
- Военные программы.
- Финансовые программы.

Верифицируеммая модель автоматных программ

- Поставщики событий и объекты управления рассматриваются как внешняя среда.
- Автомат в любой момент может совершить любой допустимый переход.

Автомат Бюхи

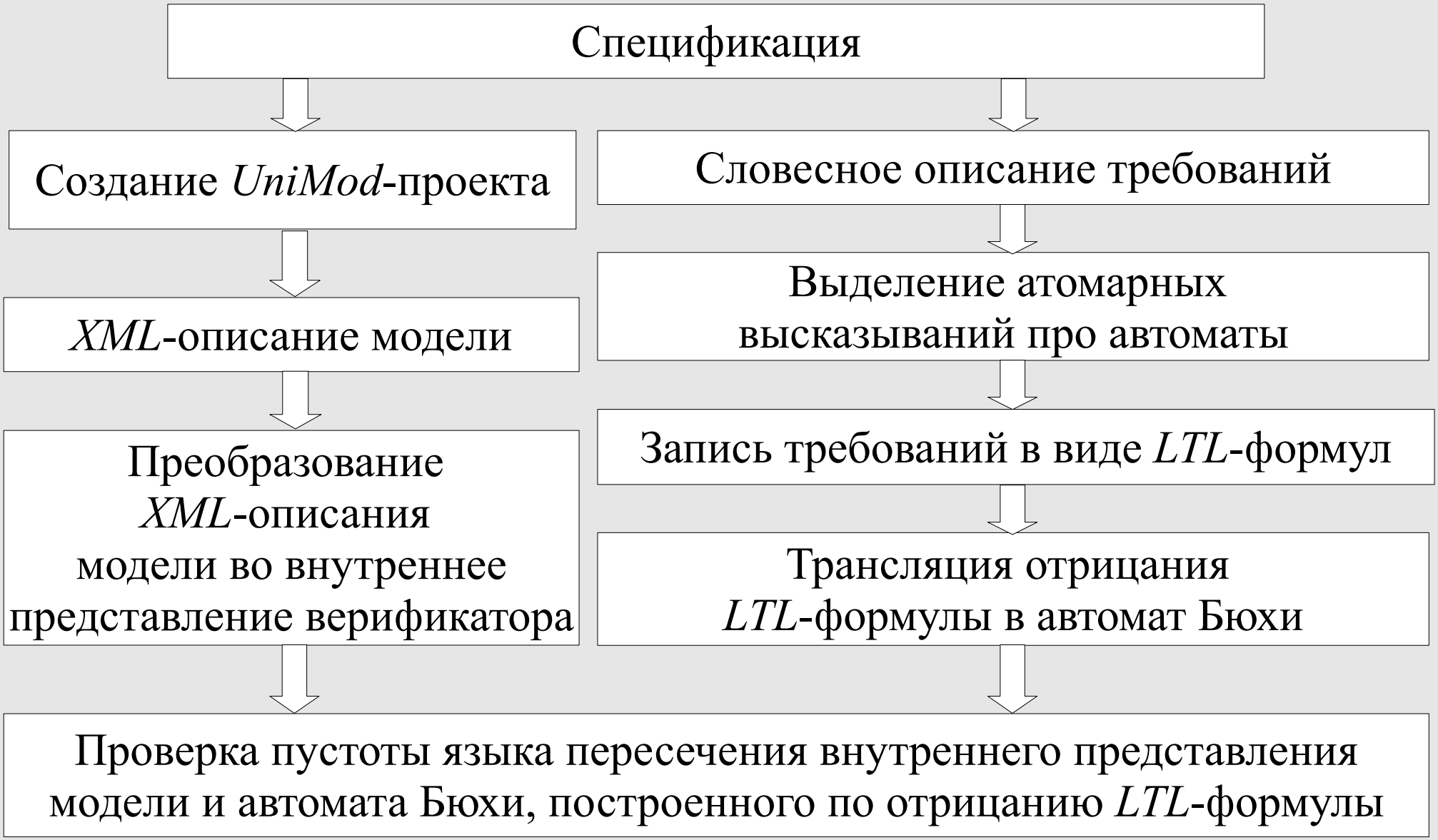
- S – конечное множество состояний;
- E – конечное множество меток переходов;
- $T \subseteq S \times E \times S$ – множество переходов;
- s_0 – начальное состояние;
- $F \subseteq S$ – множество допускающих состояний.

Язык логики LTL

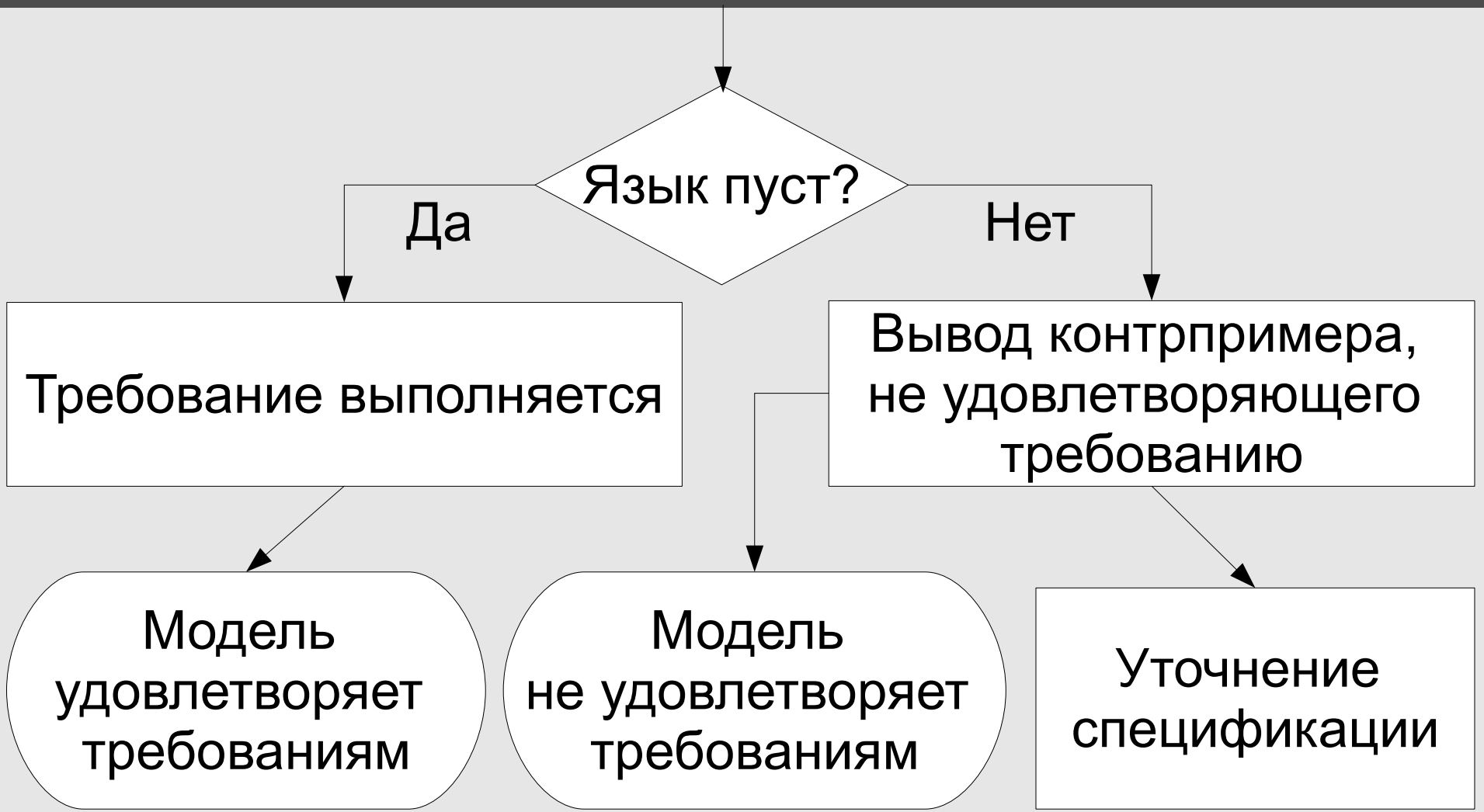
- Булевы операторы ($\neg$, $\vee$, $\wedge$).
- Пропозициональные переменные:

  $I: Prop \rightarrow \{True, False\}$.
- Временные операторы:
  - **X** (ne**X**t);
  - **F** (in the **F**uture);
  - **G** (**G**lobally in the future);
  - **U** (**U**ntil);
  - **R** (**R**elease).

Методика верификации автоматных программ 1/2



Методика верификации автоматных программ 2/2



Пример работы работы верификатора 1/4

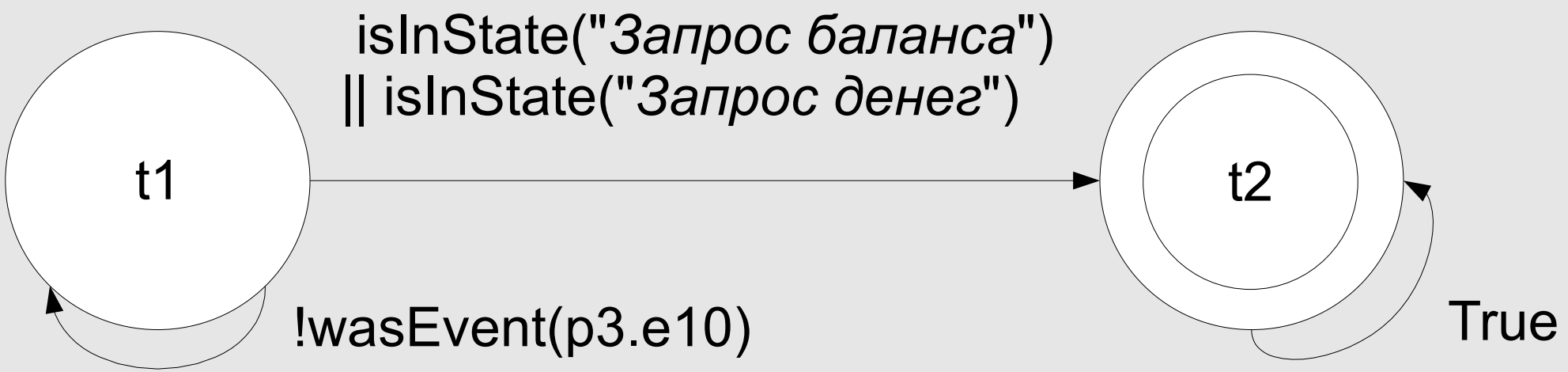
- «Пользователь не может запросить снятие наличные или запросить баланс до тех пор, пока не пройдет авторизацию».
- «Автомат *AClient* не попадет в состояние «*Запрос баланса*» или в состояние «*Запрос денег*» до тех пор, пока не произойдет событие *p3.e10*».

Пример работы работы верификатора 2/4

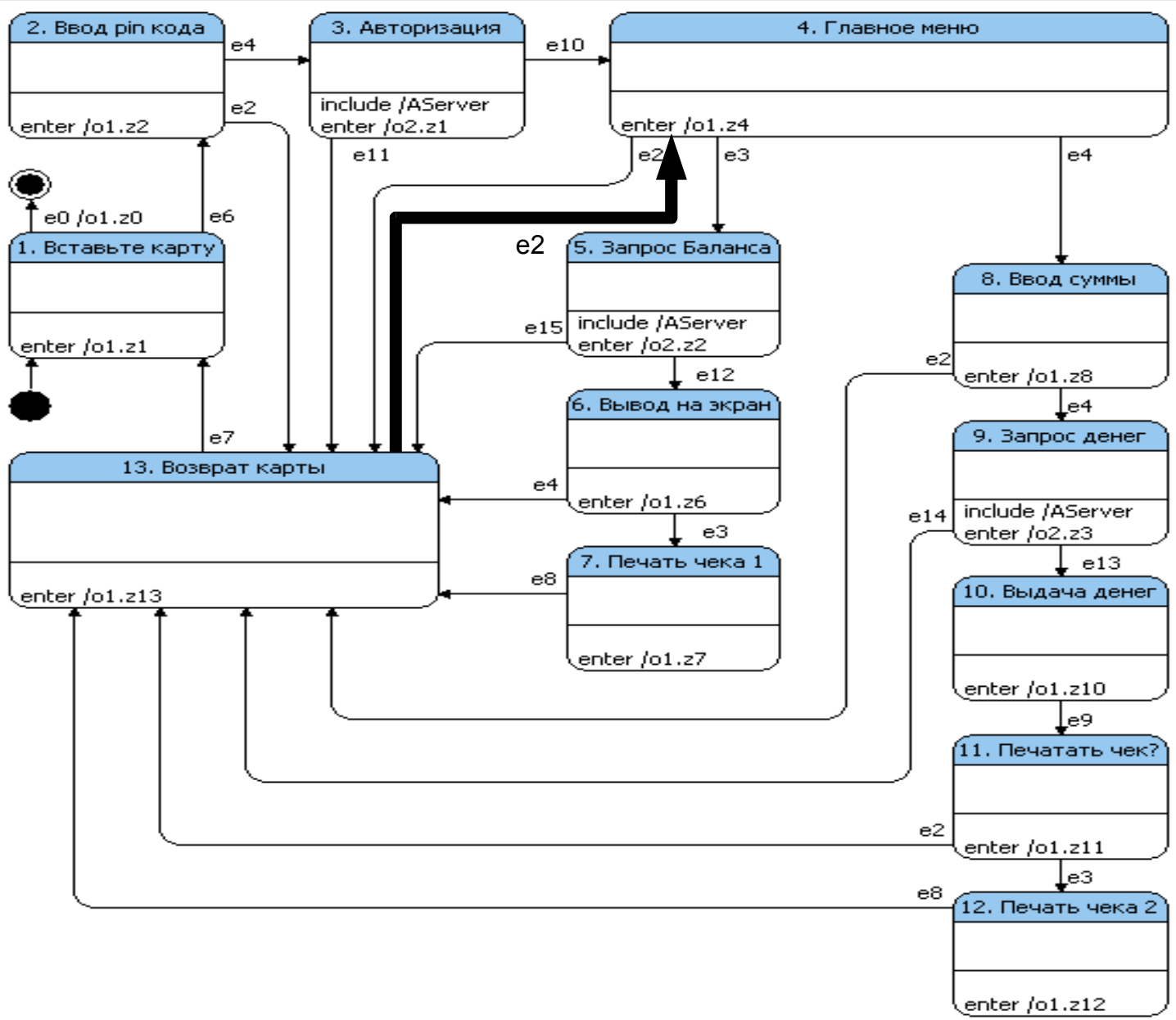
- «*wasEvent(p3.e10) R*

  (!isInState(AClient[«Запрос баланса»]) &&

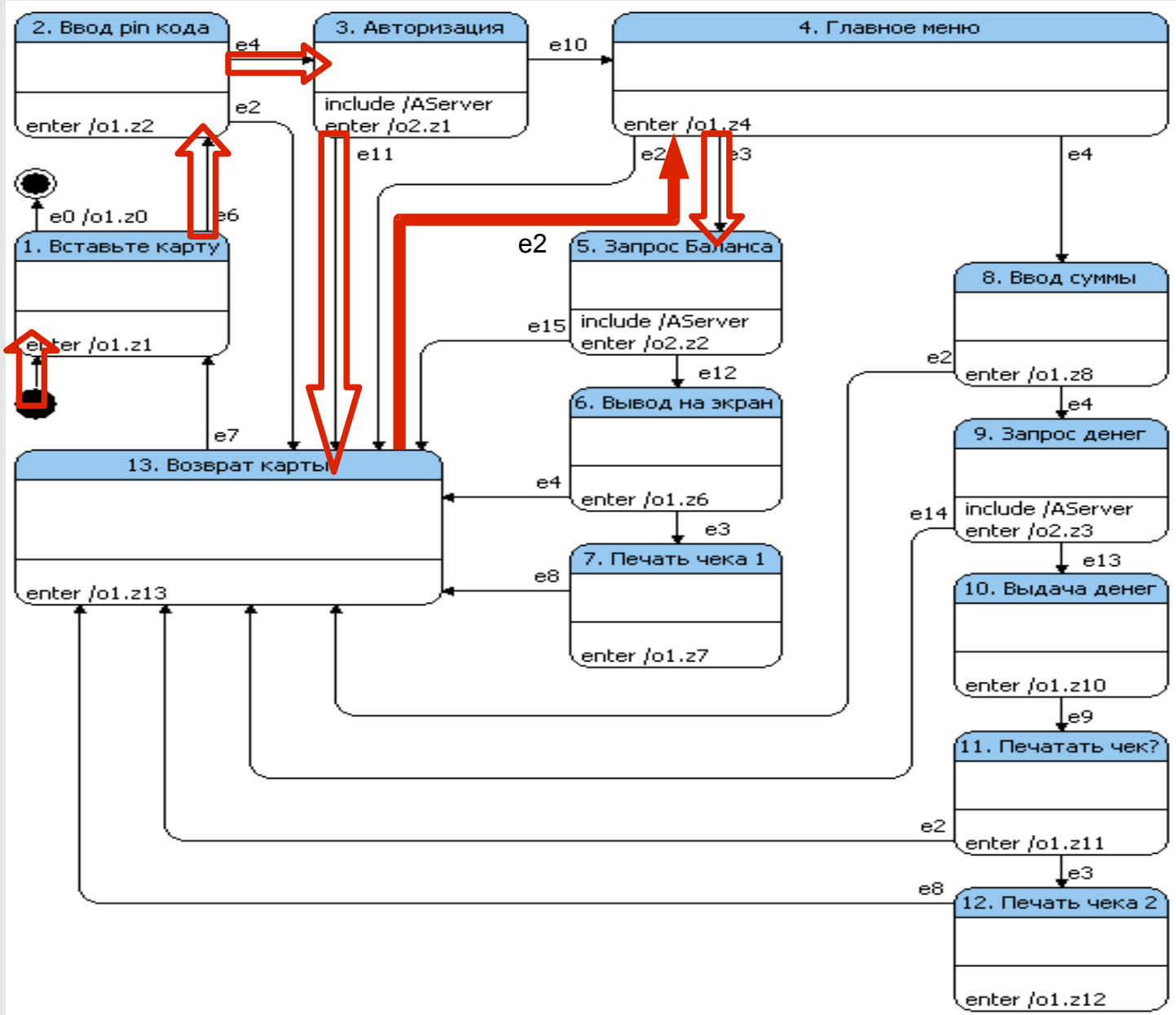
  !isInState(AClient[«Запрос денег»]))».



Пример работы работы верификатора 3/4



Пример работы работы верификатора 4/4



Распараллеливание процесса верификации 1/3

- Распараллеливание двойного обхода в глубину.
- Известно – верификатор *Spin*.
- Предлагается другая модификация алгоритма двойного обхода в глубину.

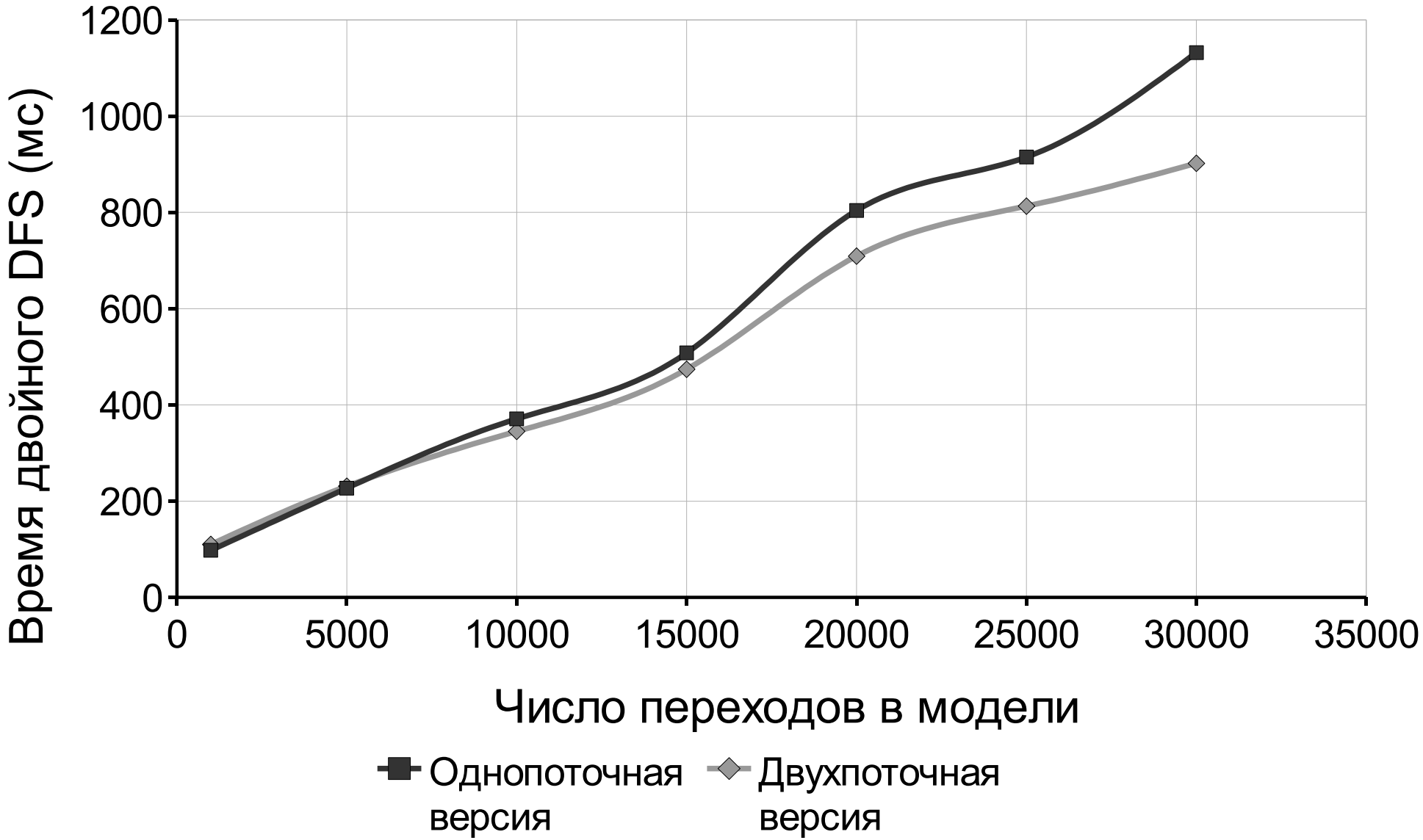
Распараллеливание процесса верификации 2/3

- Состояния пересечения автоматов обходят одновременно несколько потоков.
- Общее множество посещенных состояний.
- Стек одного потока – путь в дереве стеков от корня до листа.
- Потоки обходят в глубину непосещенные состояния.

Распараллеливание процесса верификации 3/3

- При возвращении в состояние, поток не сразу покидает его, а пытается помочь другому потоку.
- Поток обходит дерево стеков в глубину, пока не обнаружит состояние, из которого ведут переходы в непосещенные состояния.

Тестирование многопоточного верификатора



Итог работы

- **Создание верификатора *UniMod*-модели.**
- Трансляция *XML*-описания *UniMod*-модели в верифицируемую модель.
- Трансляция *LTL*-формулы в автомат Бюхи:
  - Собственный транслятор.
  - Использование *LTL2BA*.
- Проверка пустоты языка пересечения модели и отрицания *LTL*-формулы:
  - Двойной обход в глубину (*DFS*).
  - Многопоточная модификация двойного *DFS*.
- Верификация *UniMod*-проектов.

Вопросы