

Математические методы верификации схем и программ

mk.cs.msu.ru → Лекционные курсы
→ Математические методы верификации схем и программ

Блок 8

Модели Крипке

Лектор:

Подымов Владислав Васильевич

E-mail:

valdus@yandex.ru

ВМК МГУ, 2024/2025, осенний семестр

Вступление

Обычно модель в рамках метода проверки моделей устроена так:

- ▶ Моделью задаётся множество **состояний**: «слепков» системы, в которых записаны рассматриваемые особенности системы в заданные моменты времени выполнения
- ▶ Состояния могут изменяться посредством выполнения **переходов**, изменяющих текущее состояние согласно выполнению заданных **действий** системой
- ▶ Выполнение системы в неограниченном времени соответствует **вычислению** модели: бесконечной последовательности состояний, получающейся из заданного состояния выполнением переходов

Вступление

Проверка моделей применяется *в основном* для анализа систем с **конечным** числом состояний

Это один из недостатков метода, затрудняющих его широкое использование: на практике число состояний системы нередко бесконечно или конечно, но настолько велико, что можно считать его практически бесконечным

Тем не менее, существуют и важные классы систем, заведомо обладающие «разумно»-конечным числом состояний: контроллеры, драйверы, многие коммуникационные протоколы, не слишком объёмная аппаратура, ...

Вступление

Обсуждение моделей вычислительных систем начнём немного издалека, с классической головоломки про волка, козу и капусту

На левом берегу реки располагаются

волк () , коза () , капуста () и лодочник с лодкой ()

 может переправиться на противоположный берег в лодке, взяв с собой не более одного пассажира (, , )

Оставшись на берегу без ,

 может съесть , а  может съесть 

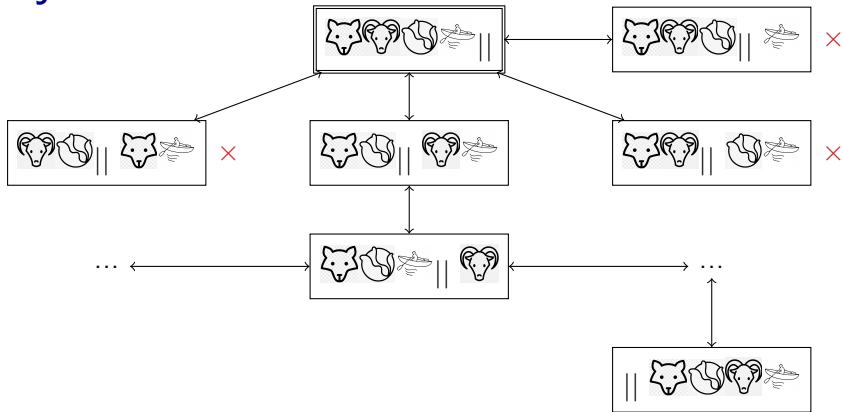
Как  может безопасно переправить ,  и  на правый берег?

Вступление

Чтобы решить эту головоломку, достаточно

- ▶ перебрать всевозможные варианты расположения  ,  ,  и  ,
получающиеся из начального расположения
согласно всевозможным действиям 
 - ▶ это **состояния** системы
- ▶ разделить состояния на “плохие” (кто-то кого-то может съесть)
и “хорошие” (остальные)
 - ▶ пометим плохие состояния символом ✗
- ▶ посмотреть, как достичь состояния “все на правом берегу”,
ни разу не встретив ✗

Вступление



Модели Крипке

Для множества X записью 2^X будем обозначать множество всех подмножеств X

Модель Крипке над множеством атомарных высказываний AP — это система $M = (S, S_0, \rightarrow, L)$, где:

- ▶ S — множество состояний
- ▶ S_0 — множество начальных состояний, $S_0 \subseteq S$
- ▶ $\rightarrow \subseteq S \times S$ — тотальное отношение переходов
- ▶ $L : S \rightarrow 2^{AP}$ — функция разметки

Тотальность отношения переходов означает, что для любого состояния s существует состояние s' , такое что $s \rightarrow s'$

Событием будем называть произвольное множество атомарных высказываний (элемент семейства 2^{AP})

Модели Крипке

$(M = (S, S_0, \rightarrow, L))$ — модель Крипке над AP)

Соотношение $L(s) = \sigma$ можно понимать так: состояние s обладает свойствами, отвечающими атомарным высказываниям из σ , и не обладает остальными свойствами, отвечающими атомарным высказываниям

Будем говорить, что модель M **конечна**, если конечны множества S и AP

Модель M представляет собой особый размеченный ориентированный граф: S — это вершины, $\rightarrow$ — это дуги, остальное — это метки вершин

В связи с этим будем применять графовые обозначения и графовую терминологию к моделям Крипке

Модели Крипке

$(M = (S, S_0, \rightarrow, L))$ — модель Крипке над AP)

Путь в модели Крипке, исходящий из начального состояния, будем называть **начальным**

Бесконечный начальный путь будем наывать **вычислением** модели

Трассой будем называть бесконечную последовательность событий

Иногда будут представлять интерес и конечные последовательности событий — будем называть их **конечными трассами**, или просто трассами, если конечность следует из контекста

Трассой пути $s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow \dots$ в модели Крипке будем называть трассу, состоящую из событий, помечающих состояния этого пути:

$$L(s_1), L(s_2), L(s_3), \dots$$