

Применение алгоритмов АФП в задачах тестирования ПО

Тестирование

- Процесс исследования, испытания ПО
- 2 цели:
 - Продемонстрировать соответствие программы требованиям
 - Выявить ситуации, где поведение неправильное, нежелательное или не соответствует спецификации

Объекты тестирования

- Функциональное
- Производительности
- Конфигурации
- Юзабилити
- Интерфейса
- Безопасности
- Локализации

Исполнимость кода

- Статическое
 - Код не запускается
 - Анализ вручную
 - Специальные инструменты статического анализа
- Динамическое

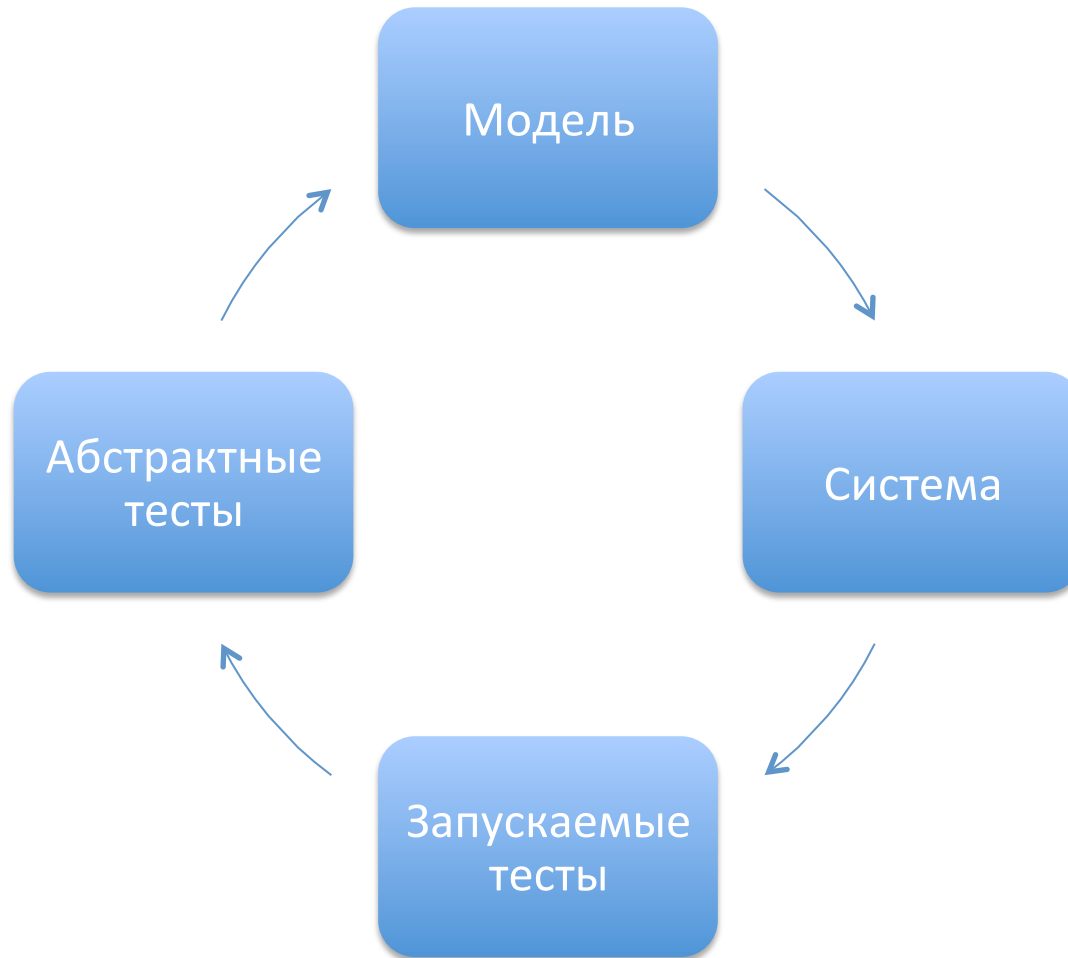
Уровни знания системы

- Черный ящик
 - Нет знания о внутреннем устройстве исследуемой системы
- Белый ящик
 - С точки зрения компилятора того языка, на котором писалась программа
 - Мутационное тестирование
- Серый ящик

Виды тестирования

- Структуры
 - Программисты/инженеры
- Поведения
 - Инженеры по контролю качества
- «На живую» (alpha/beta,...)
 - Аналитики / пользователи

Model-based



- Требования

- Модель

- Последовательность тестов

- Оракул

- Реализация

- » Вердикт: тест проходит или проваливается

Модели

- Есть направление инжиниринг на основе моделей (Model Driven Engineering)
 - Можно получить из существующих систем
 - Можно строить перед реализацией
 - Можно - параллельно с самой системой

Варианты реализации

- **Онлайн**
 - Подключается непосредственно к системе, которую тестирует
- **Оффлайн**
 - Генерирует тестовые случаи, которые могут быть использованы впоследствии

Возможные варианты реализации

- Конечные автоматы
- Доказательство теорем
- Программирование в условиях ограничений (constraint programming)
- На основе Марковских цепей
- Моделирование пространства входных значений

Анализ формальных понятий

- G – множество объектов
- M – множество признаков
- I – бинарное отношение между G и M ,
физический смысл: объект обладает
признаком

$$A' = \{m \in M \mid gIm \text{ for } g \in A\},$$

$$B' = \{g \in G \mid gIm \text{ for all } m \in B\}$$

Формальные понятия и импликации

- Если есть все признаки из A , то есть и все признаки из B :

$$A, B \subseteq M, A \rightarrow B, \text{ если } A' \subseteq B'$$

- Правила Армстронга:

$$\overline{X \rightarrow X}$$

$$X \rightarrow Y$$

$$\overline{X \cup Z \rightarrow Y}$$

$$X \rightarrow Y, Y \cup Z \rightarrow W$$

$$\overline{X \cup Z \rightarrow W}$$

Исследование признаков

- Интерактивная процедура исследования предметной области
- Поддерживает две сущности: множество импликаций и множество контрпримеров
- Может быть начат с произвольного начального состояния для обоих множеств

Пример

- Пусть есть программа, которая на вход получает одно число и выдает признаки, которыми оно обладает.
- Поддерживаемые признаки: `even`, `factorial`, `divided_by_three`, `odd`, `prime`

Пример

- => even, factorial, divided by three, odd, prime

Пример

G\M	even	factorial	Divided_by_ three	odd	prime
2	X	X			

Пример

- \Rightarrow even, factorial, prime

Пример

G\M	even	factorial	Divided_by_ three	odd	prime
2	X	X			X
5				X	X

Пример

- \Rightarrow prime

Пример

G\M	even	factorial	Divided_by_ three	odd	prime
2	X	X			X
5				X	X
6	X	X	X		

Пример

- factorial \Rightarrow even

Пример

G\M	even	factorial	Divided_by_ three	odd	prime
2	X	X			X
5				X	X
6	X	X	X		
1		X		X	X

Пример

- $\text{odd} \Rightarrow \text{prime}$

Пример

G\M	even	factorial	Divided_by_ three	odd	prime
2	X	X			X
5				X	X
6	X	X	X		
1		X		X	X
9			X	X	

Пример

- factorial, odd \Rightarrow prime
- Да, 1

Пример

- factorial, divided by three => even
- Да, первый факториал, делящийся на 3, уже делится на 2

Пример

- prime, divided by three \Rightarrow even, factorial, odd

Пример

G\M	even	factorial	Divided_by_three	odd	prime
2	X	X			X
5				X	X
6	X	X	X		
1		X		X	X
9			X	X	
3			X	X	X

Пример

- prime, divided by three \Rightarrow odd
- Да: 3

Пример

- `even => factorial`

Пример

G\M	even	factorial	Divided_by_ three	odd	prime
2	X	X			X
5				X	X
6	X	X	X		
1		X		X	X
9			X	X	
3			X	X	X
8	X				

Пример

- even, odd => factorial, prime, divided by three
- В силу ложности посылки

Пример

- even, divided by three => factorial

Пример

G\M	even	factorial	Divided_by_ three	odd	prime
2	X	X			X
5				X	X
6	X	X	X		
1		X		X	X
9			X	X	
3			X	X	X
8	X				
12	X		X		

Пример

- even, prime => factorial
- Да: 2
- Процесс окончен

Пример

G\M	even	factorial	Divided_by_ three	odd	prime
2	X	X			X
5				X	X
6	X	X	X		
1		X		X	X
9			X	X	
3			X	X	X
8	X				
12	X		X		

Пример

- Список подтвержденных импликаций:
- factorial, odd \rightarrow prime
- factorial, divided by three \rightarrow even
- prime, divided by three \rightarrow odd
- even, odd \rightarrow factorial, prime, divided by three
- even, prime \rightarrow factorial

МВТ подход

- Even: 1, 0
- Factorial: 1, 0
- Divided_by_three: 1, 0
- Odd: 1, 0
- Prime: 1, 0

МВТ: без условий

even	factorial	Divided_by_thr ee	odd	prime
1	1	0	0	1
0	0	1	1	0
1	0	1	1	1
0	1	0	1	0
1	0	1	0	0
0	1	1	0	1
0	0	0	0	0

Согласованная модель

- Event: 1, 0
- Factorial: 1, 0
- Divs3: 1, 0
- Odd: 1, 0
- Prime: 1, 0
- Result: 12, 9, 6, 3, 2, 1

Импlications

- • IF [Even] = 1 THEN [Odd] = 0 ELSE [Odd] = 1;
- • IF [Odd] = 1 AND [Factorial] = 1 THEN [Result] = 1;
- • IF [Even] = 1 AND [Prime] = 1 THEN [Result] = 2;
- • IF [Divs3] = 1 AND [Prime] = 1 THEN [Result] = 3;
- • IF [Divs3] = 1 AND [Even] = 1 THEN [Result] IN 6, 12;
- • IF [Divs3]=1 AND [Odd]=1 AND[Prime]=0 THEN [Result] = 9;
- • IF [Even] = 1 AND [Factorial] = 1 AND [Divs3] = 0 THEN [Result] = 2;

Зависимости для данных

- IF [Result] = 2 THEN [Even] = 1 AND [Factorial] = 1 AND [Divs3] = 0 AND [Odd] = 0 AND [Prime] = 1;
- IF [Result] = 3 THEN [Even] = 0 AND [Factorial] = 0 AND [Divs3] = 1 AND [Odd] = 1 AND [Prime] = 1;
- IF [Result] = 6 THEN [Even] = 1 AND [Factorial] = 1 AND [Divs3] = 1 AND [Odd] = 0 AND [Prime] = 0;
- IF [Result] = 9 THEN [Even] = 0 AND [Factorial] = 0 AND [Divs3] = 1 AND [Odd] = 1 AND [Prime] = 0;
- IF [Result] = 12 THEN [Even] = 1 AND [Factorial] = 0 AND [Divs3] = 1 AND [Odd] = 0 AND [Prime] = 0;

Результат генерации тест-кейсов

event	factorial	Divided_by_ three	odd	prime	Result
0	0	1	1	1	3
1	1	1	0	0	6
0	0	1	1	0	9
1	0	1	0	0	12
1	1	0	0	1	2
0	1	0	1	1	1

Плюсы

- На выходе 2 артефакта:
 - Тестовые случаи
 - Описание зависимостей между признаками в виде импликаций
- Поддерживает возможность расширения множества признаков:
 - У существующих объектов его надо заполнить верно, или в случае делимости, считать, что у всех имеющихся данный признак отсутствует

Пример 2

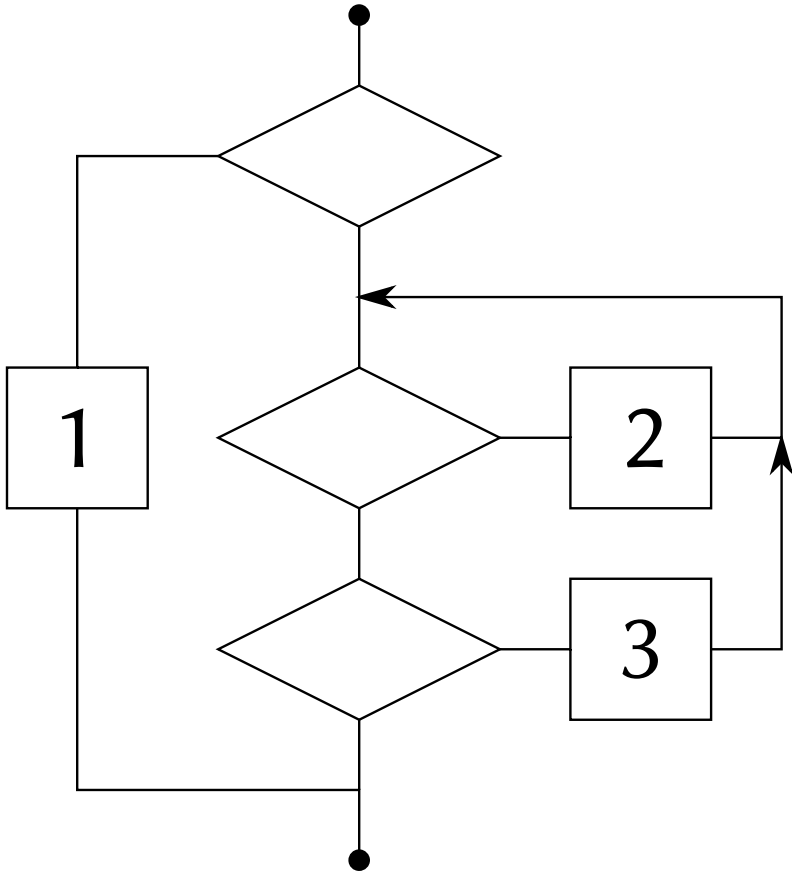
- Пример 1 исследовал зависимости для значений выхода, можно делать с учетом входа
- Интернет-реклама: есть внешние ограничения – например, рекламодатель задает регион, где хочет, чтобы показывался баннер
- Выход: показ баннера
- Вход: параметры, влияющие на геотаргетинг (запрос, заголовки, явное указание региона)

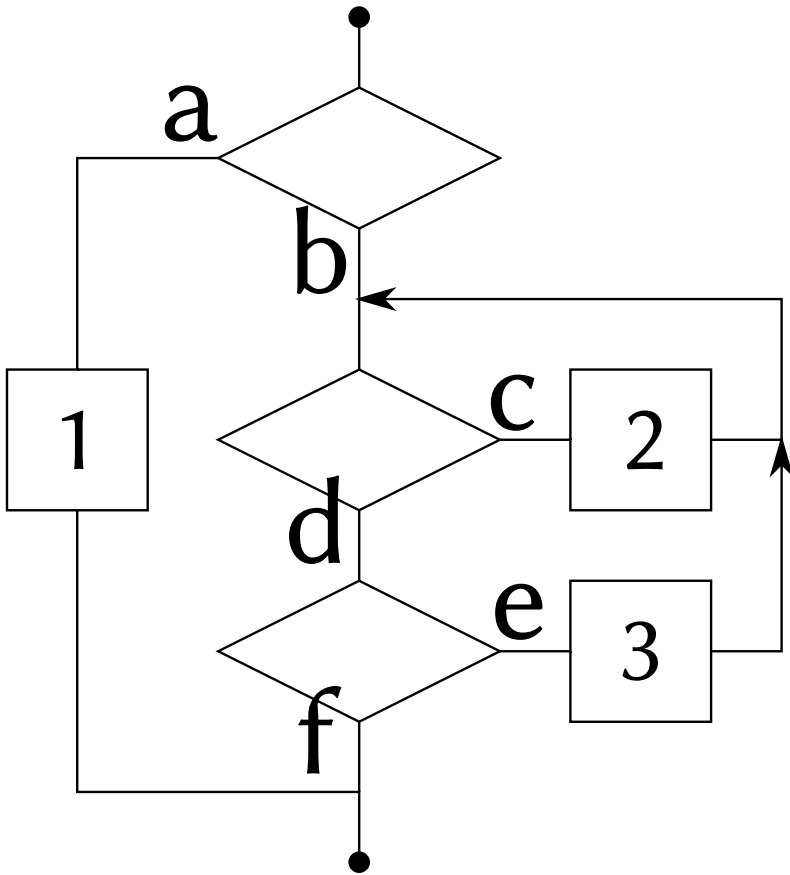
Признаки:

- IsQueryPresent(IQP), IsQueryValid(IQV),
- IsSqPresent(ISP), IsSqValid(ISV),
- IsIpPresent(IIP), IsIpValid(IIV),
- UserRegion == BannerRegion(=),
- UserRegion < BannerRegion(<),
- UserRegion > BannerRegion(>),
- IsBannerShown(IBS)

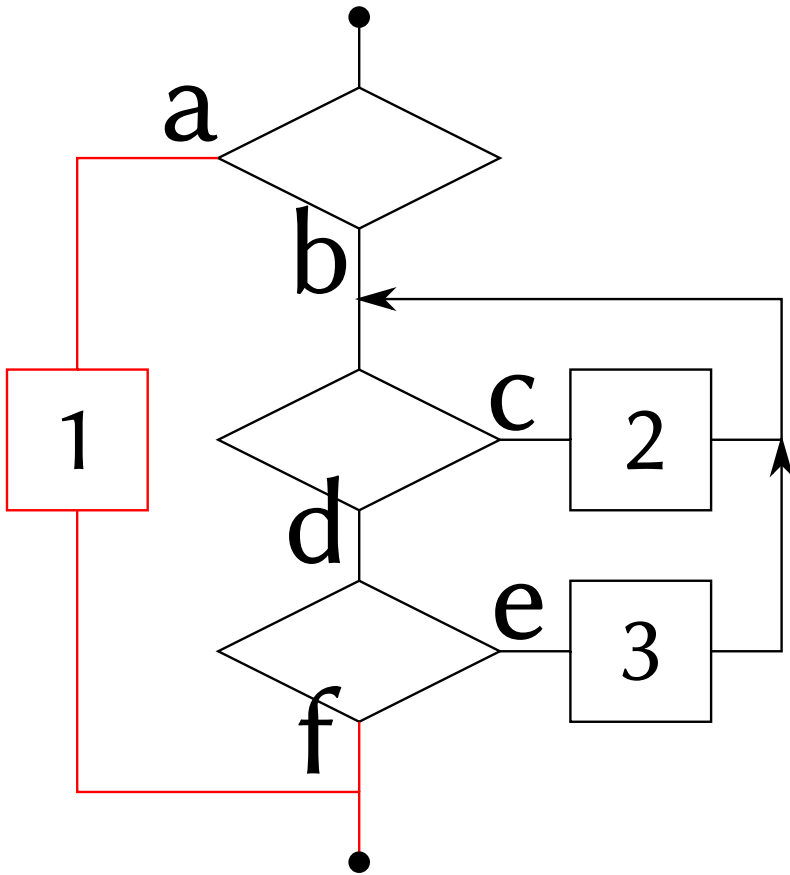
White-box

- Можно использовать attribute exploration для покрытия веток кода



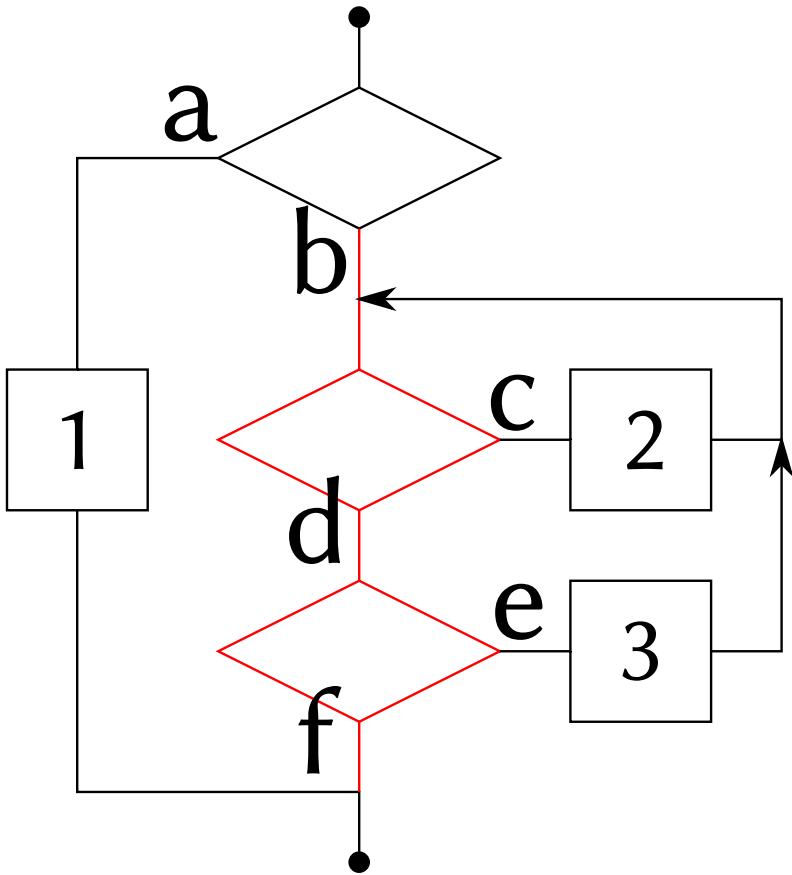


a b c d e f

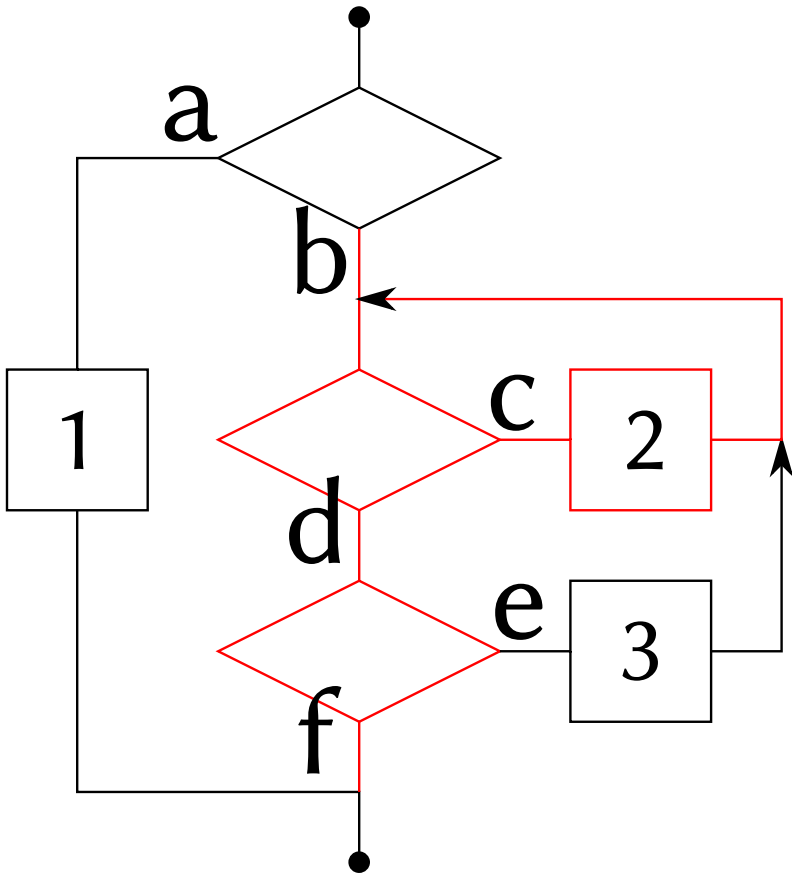


a b c d e f

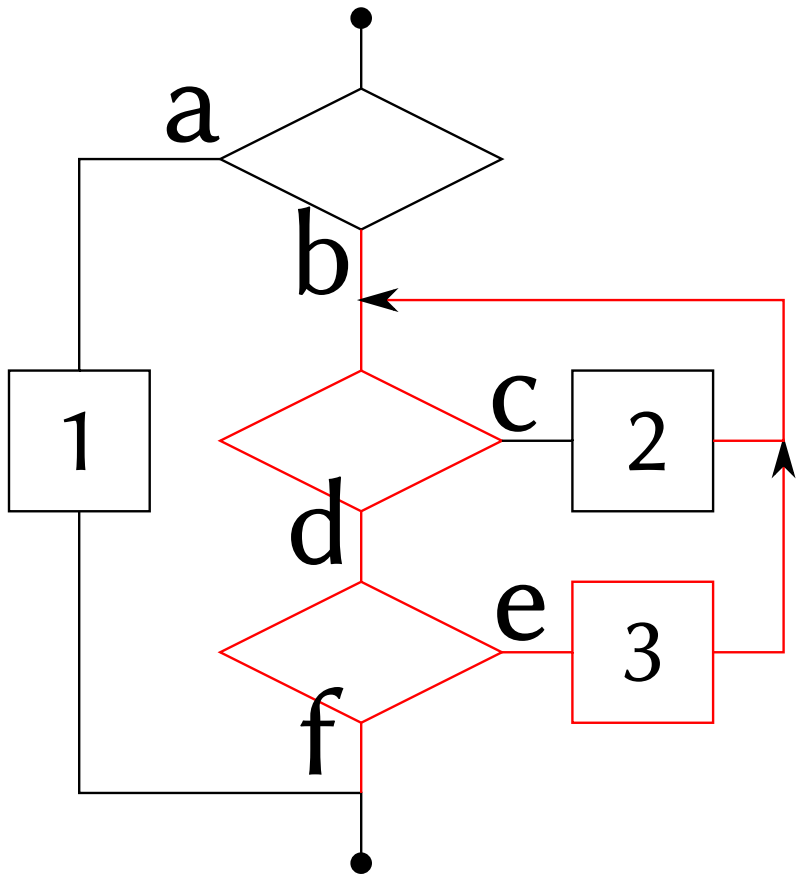
X



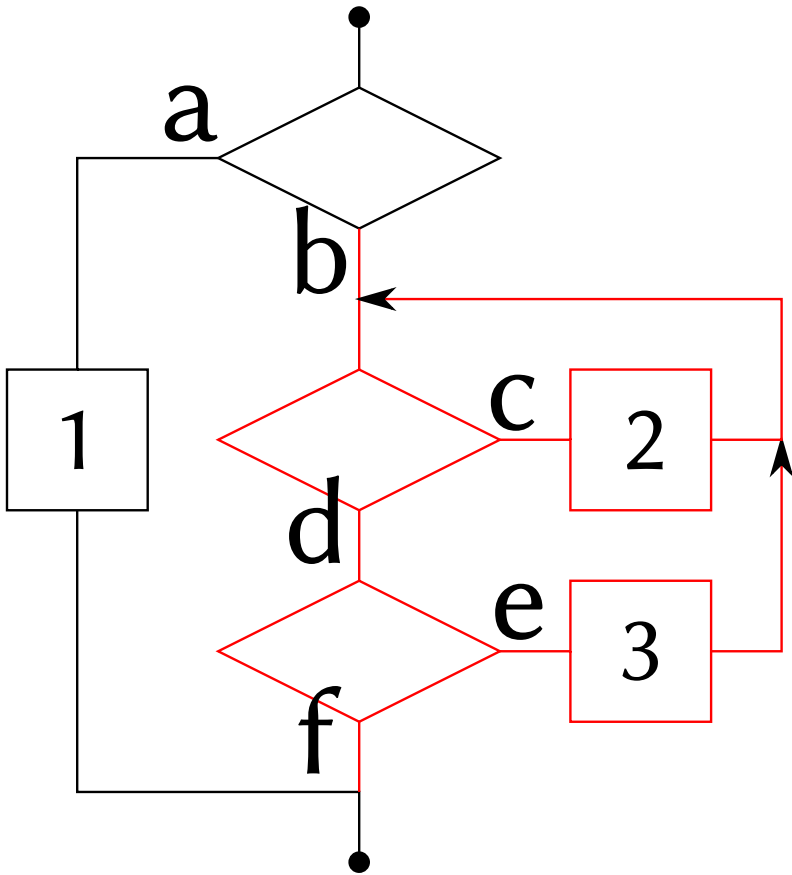
a	b	c	d	e	f
×					
	×		×		×



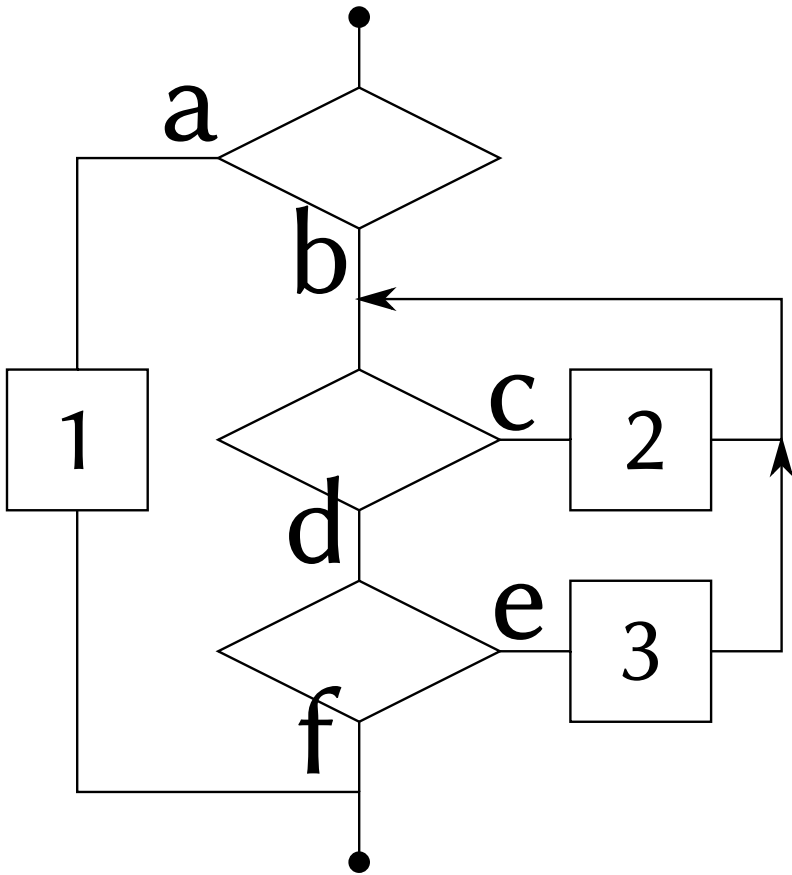
a	b	c	d	e	f
×					
	×		×		×
	×	×	×		×



a	b	c	d	e	f
×					
	×		×		×
	×	×	×		×
	×		×	×	×



a	b	c	d	e	f
×					
	×		×		×
	×	×	×		×
	×		×	×	×
	×	×	×	×	×



a	b	c	d	e	f
×					
	×		×		×
	×	×	×		×
	×		×	×	×
	×	×	×	×	×

a b c d e f

X

X

X

X

X

X

X

X

X

X

X

X

X

X

X

X

X

$\emptyset \rightarrow a, b, c, d, e, f$

$\emptyset \rightarrow a$

$b, a \rightarrow c, d, e, f$

$c \rightarrow a, b, d, e, f$

$e \rightarrow a, b, c, d, f$

$f, e, d, c, b \rightarrow a$

$b \rightarrow d, f$

$c \rightarrow b, d, f$

$d \rightarrow b, f$

$e \rightarrow b, d, f$

$f \rightarrow b, d$

$f, d, b, a \rightarrow c, e$

...

a b c d e f

~~$\emptyset \rightarrow a, b, c, d, e, f$~~

$\emptyset \rightarrow a$

$b, a \rightarrow c, d, e, f$

$c \rightarrow a, b, d, e, f$

$e \rightarrow a, b, c, d, f$

$f, e, d, c, b \rightarrow a$

$b \rightarrow d, f$

$c \rightarrow b, d, f$

$d \rightarrow b, f$

$e \rightarrow b, d, f$

$f \rightarrow b, d$

$f, d, b, a \rightarrow c, e$

...

a b c d e f
X

~~$\emptyset \rightarrow a, b, c, d, e, f$~~

~~$\emptyset \rightarrow a$~~

$b, a \rightarrow c, d, e, f$

~~$c \rightarrow a, b, d, e, f$~~

$e \rightarrow a, b, c, d, f$

$f, e, d, c, b \rightarrow a$

$b \rightarrow d, f$

$c \rightarrow b, d, f$

$d \rightarrow b, f$

$e \rightarrow b, d, f$

$f \rightarrow b, d$

$f, d, b, a \rightarrow c, e$

...

a	b	c	d	e	f
X					
	X		X		X
	X	X	X		X

~~$\emptyset \rightarrow a, b, c, d, e, f$~~

~~$\emptyset \rightarrow a$~~

$b, a \rightarrow c, d, e, f$

~~$c \rightarrow a, b, d, e, f$~~

~~$e \rightarrow a, b, c, d, f$~~

$f, e, d, c, b \rightarrow a$

$b \rightarrow d, f$

$c \rightarrow b, d, f$

$d \rightarrow b, f$

$e \rightarrow b, d, f$

$f \rightarrow b, d$

$f, d, b, a \rightarrow c, e$

...

<u>a</u>	<u>b</u>	<u>c</u>	<u>d</u>	<u>e</u>	<u>f</u>
×					
	×		×		×
	×	×	×		×
	×		×	×	×

~~$\emptyset \rightarrow a, b, c, d, e, f$~~

~~$\emptyset \rightarrow a$~~

$b, a \rightarrow c, d, e, f$

~~$c \rightarrow a, b, d, e, f$~~

~~$e \rightarrow a, b, c, d, f$~~

~~$f, e, d, c, b \rightarrow a$~~

$b \rightarrow d, f$

$c \rightarrow b, d, f$

$d \rightarrow b, f$

$e \rightarrow b, d, f$

$f \rightarrow b, d$

$f, d, b, a \rightarrow c, e$

...

a	b	c	d	e	f
X					
	X		X		X
	X	X	X		X
	X		X	X	X
	X	X	X	X	X

~~$\emptyset \rightarrow a, b, c, d, e, f$~~

~~$\emptyset \rightarrow a$~~

3 $b, a \rightarrow c, d, e, f$

~~$c \rightarrow a, b, d, e, f$~~

~~$e \rightarrow a, b, c, d, f$~~

~~$f, e, d, c, b \rightarrow a$~~

1 $b \rightarrow d, f$

$c \rightarrow b, d, f$

$d \rightarrow b, f$

$e \rightarrow b, d, f$

$f \rightarrow b, d$

2 $f, d, b, a \rightarrow c, e$

...

$1 \wedge 2 \rightarrow 3$

a	b	c	d	e	f
×					
	×		×		×
	×	×	×		×
	×		×	×	×
	×	×	×	×	×

$b \rightarrow d, f$

$c \rightarrow b, d, f$

$d \rightarrow b, f$

$e \rightarrow b, d, f$

$f \rightarrow b, d$

$f, d, b, a \rightarrow c, e$

<u>a</u>	<u>b</u>	<u>c</u>	<u>d</u>	<u>e</u>	<u>f</u>
X					
	X		X		X
	X	X	X		X
	X		X	X	X
	X	X	X	X	X

Отрицания атрибутов

a b c

×

$a \rightarrow b, c$

$c \rightarrow a, b$

a b c

○ ○ ○

○ × ○

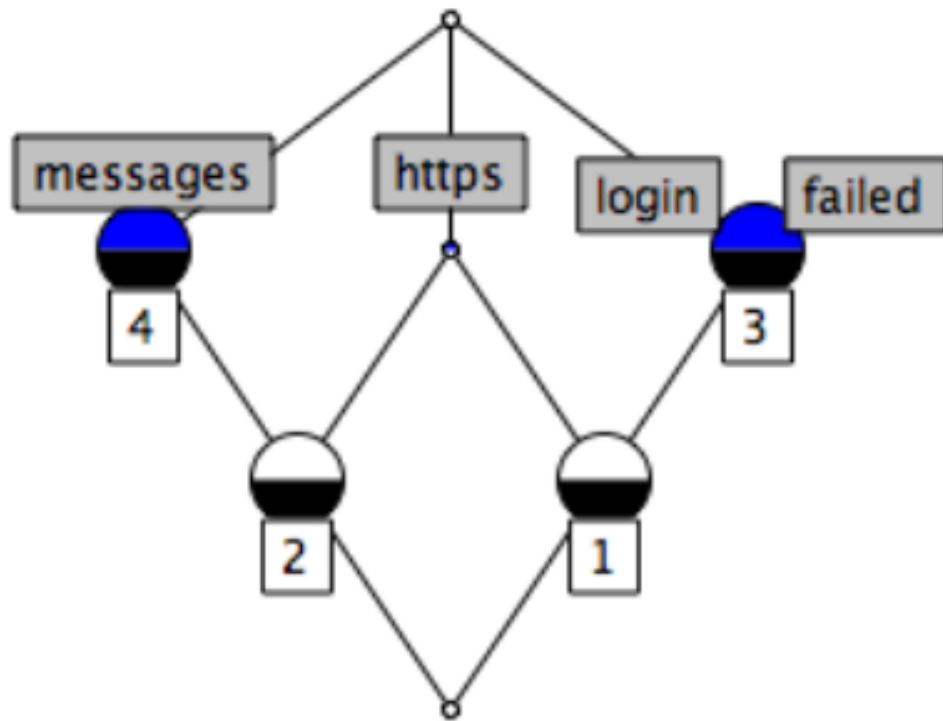
$\emptyset \rightarrow \neg a, \neg c$

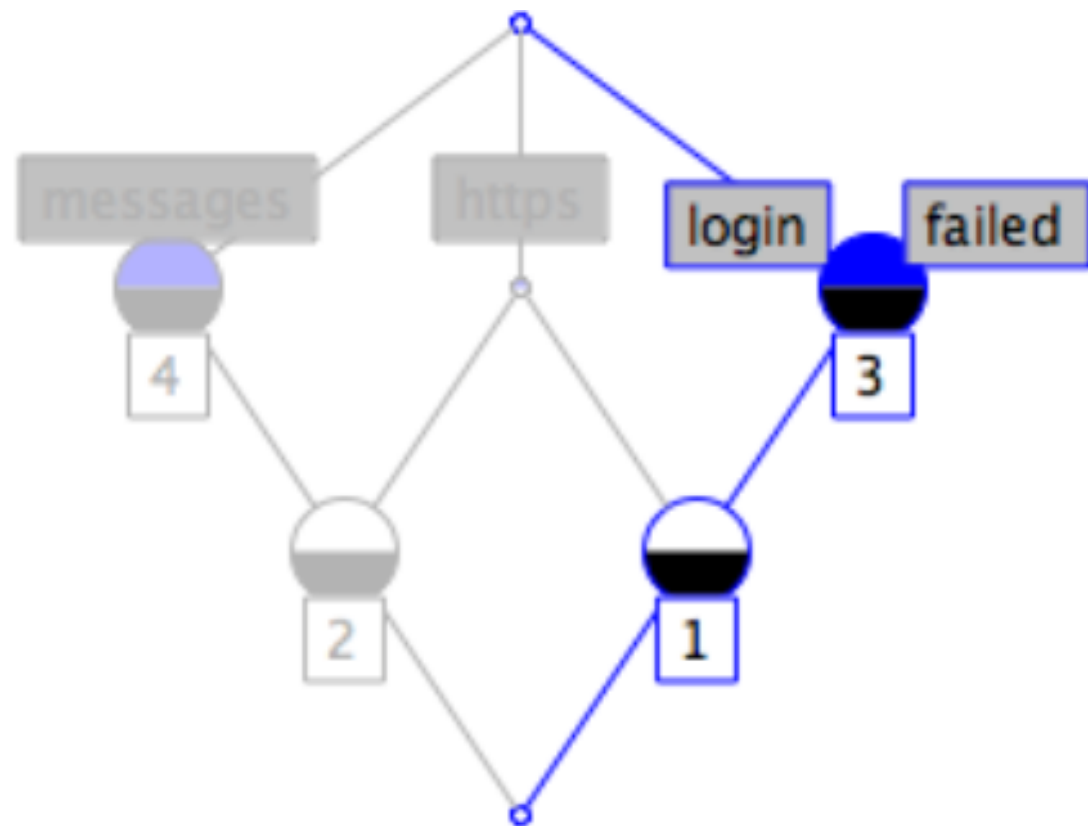
Использование решеток

- АФП могут быть частично упорядочены по операции вложения над содержаниями/поглощения над объемами
- Все формальные понятия образуют решетку
- Анализ результатов прогона тестов

Пример использования решетки

G\M	failed	https	login	messages
1	X	X	X	
2		X		X
3	X		X	
4				X





Применение решеток

- Анализ айсберга решетки для упавших тестов соответствует поиску наиболее общих описаний таких тестов
- В больших системах может быть полезно исследовать схожую функциональность – построение не древовидной иерархии частей тестируемой системы

Future work

- Полноценное использование узорных понятий
- Фреймворк для построения гипотез о причинах падения тестов
- Attribute Exploration в терминах не только булевых значений