CS 3100, Models of Computation, Spring 20, Lec 10

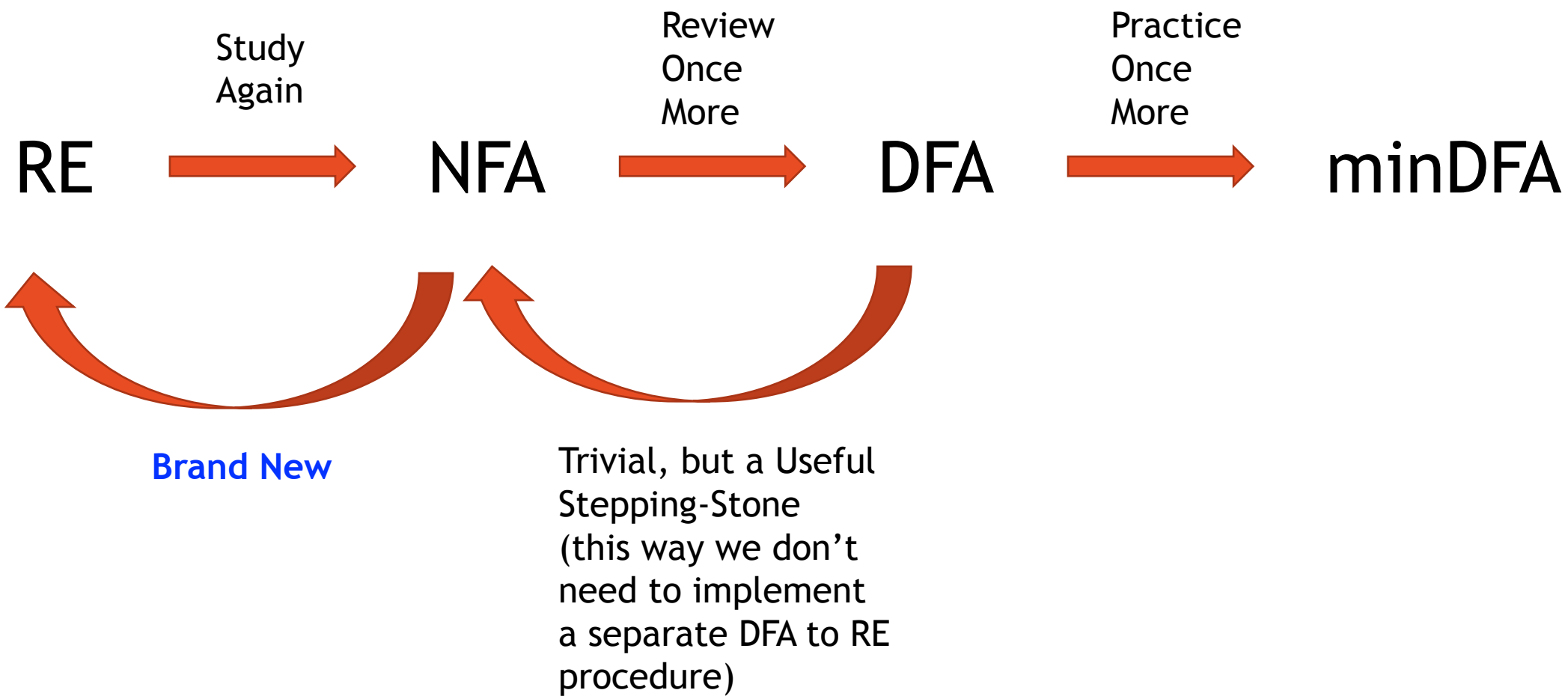
Ganesh Gopalakrishnan
School of Computing
University of Utah
Salt Lake City, UT 84112

bit.ly/3100s20Syllabus

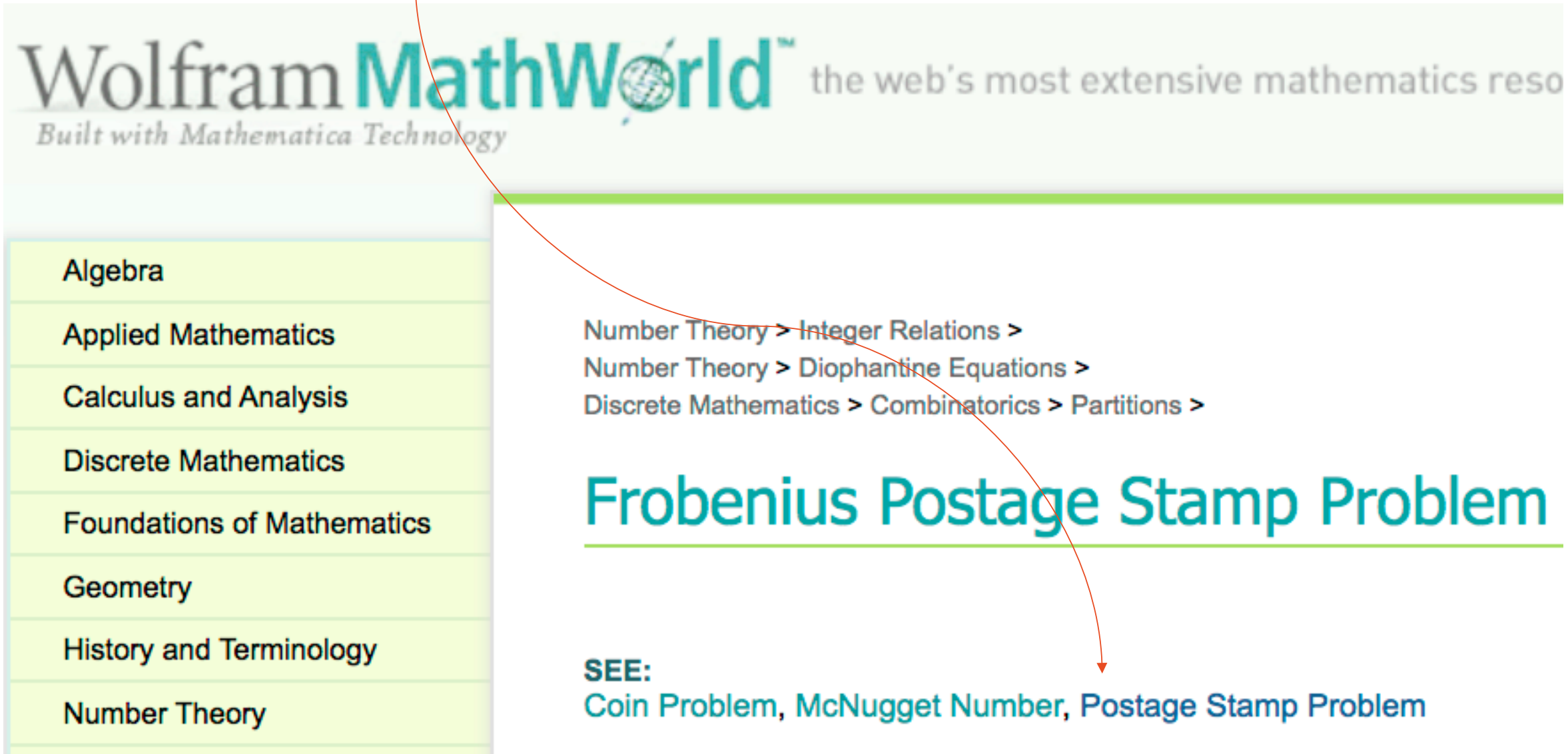
Lecture 10, covering Ch 1-9 ☺

- Walk the Kleene-Pipeline
- The Postage-Stamp Problem

Walk the Kleene-Pipeline



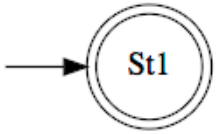
The Postage-Stamp Problem



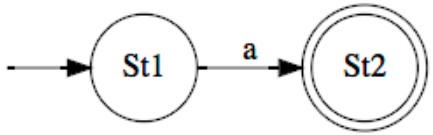
RE → NFA examples

```
1 dotObj_nfa(re2nfa("''"))
```

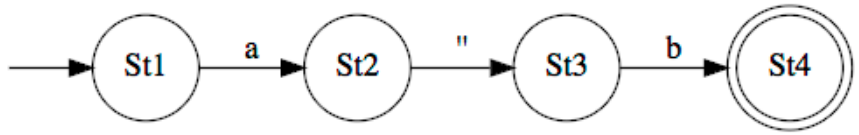
```
Generating LALR tables
```



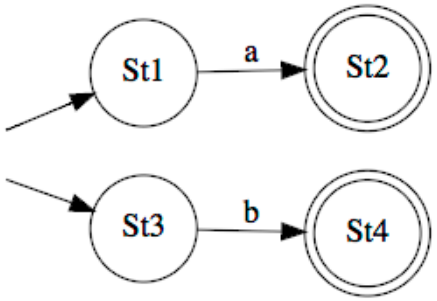

```
1 dotObj_nfa(re2nfa("a"))
```



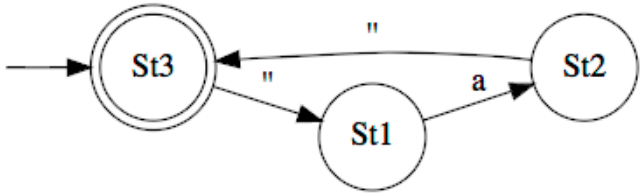

```
1 dotObj_nfa(re2nfa("ab"))
```




```
1 dotObj_nfa(re2nfa("a+b"))
```




```
1 dotObj_nfa(re2nfa("a*"))
```



What are RE?

- Epsilon
- a in Sigma
- If R1 and R2 are RE, then R1 + R2 is an RE
- If R1 and R2 are RE, then R1 R2 is an RE
- If R is an RE, then (R) is an RE
- If R is an RE, then R* is an RE
- Nothing else is an RE

Cover RE → NFA for each case

- Epsilon
- a in Sigma
- If R1 and R2 are RE, then R1 + R2 is an RE
- If R1 and R2 are RE, then R1 R2 is an RE
- If R is an RE, then (R) is an RE
- If R is an RE, then R* is an RE
- Nothing else is an RE

Cover RE → NFA for each case

- Epsilon
- a in Sigma
- If R1 and R2 are RE, then R1 + R2 is an RE
- If R1 and R2 are RE, then R1 R2 is an RE
- If R is an RE, then (R) is an RE
- If R is an RE, then R* is an RE
- Nothing else is an RE

Cover RE → NFA for each case

- Epsilon
- a in Sigma
- If R1 and R2 are RE, then R1 + R2 is an RE
- If R1 and R2 are RE, then R1 R2 is an RE
- If R is an RE, then (R) is an RE
- If R is an RE, then R* is an RE
- Nothing else is an RE

Cover RE → NFA for each case

- Epsilon
- a in Sigma
- If R1 and R2 are RE, then R1 + R2 is an RE
- If R1 and R2 are RE, then R1 R2 is an RE
- If R is an RE, then (R) is an RE
- If R is an RE, then R* is an RE
- Nothing else is an RE

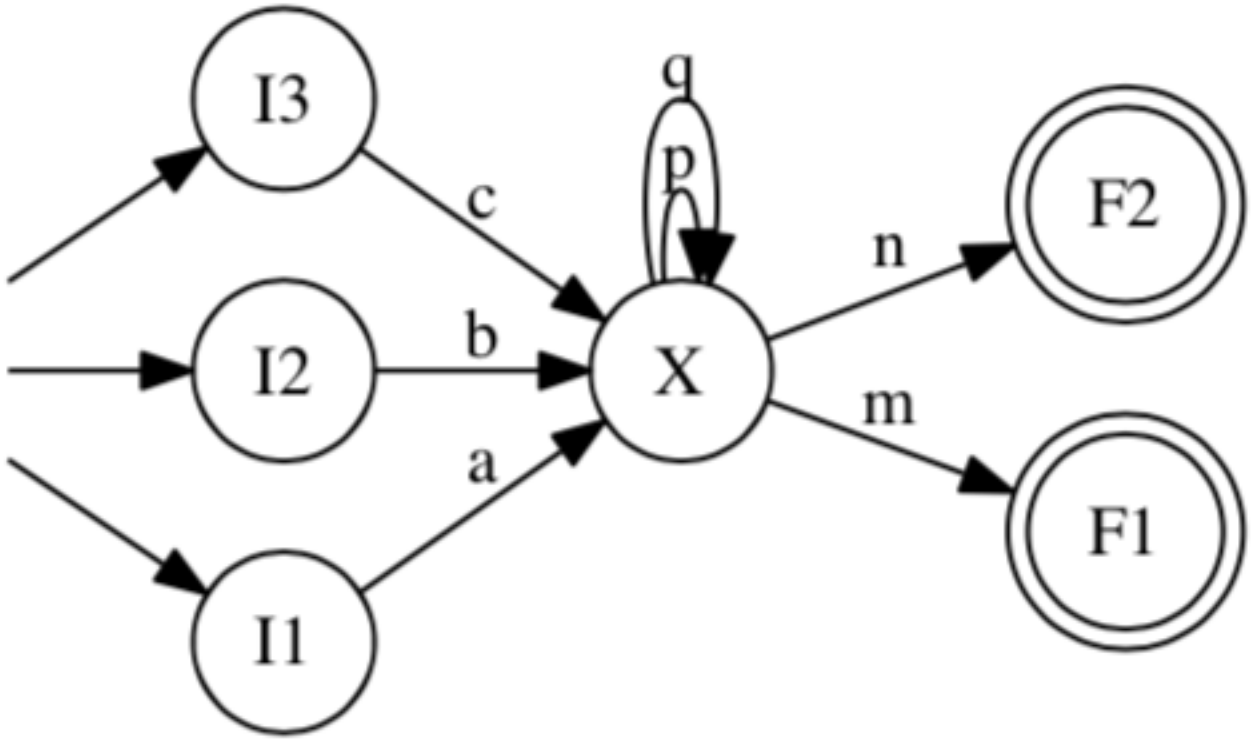
Cover RE → NFA for each case

- Epsilon
- a in Sigma
- If R1 and R2 are RE, then R1 + R2 is an RE
- If R1 and R2 are RE, then R1 R2 is an RE
- If R is an RE, then (R) is an RE
- If R is an RE, then R* is an RE
- Nothing else is an RE

Cover RE → NFA for each case

- Epsilon
- a in Sigma
- If R1 and R2 are RE, then R1 + R2 is an RE
- If R1 and R2 are RE, then R1 R2 is an RE
- If R is an RE, then (R) is an RE
- If R is an RE, then R* is an RE
- Nothing else is an RE

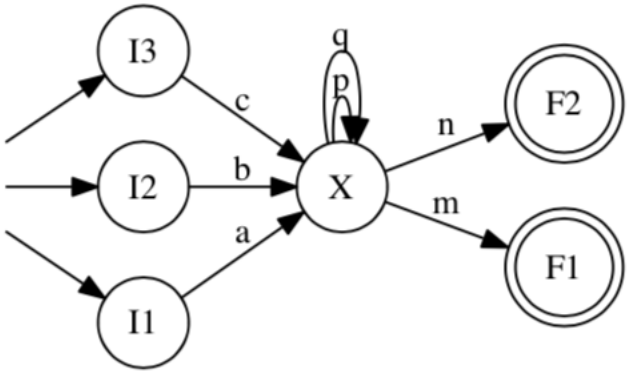
NFA to RE conversion



What is the language of this NFA?

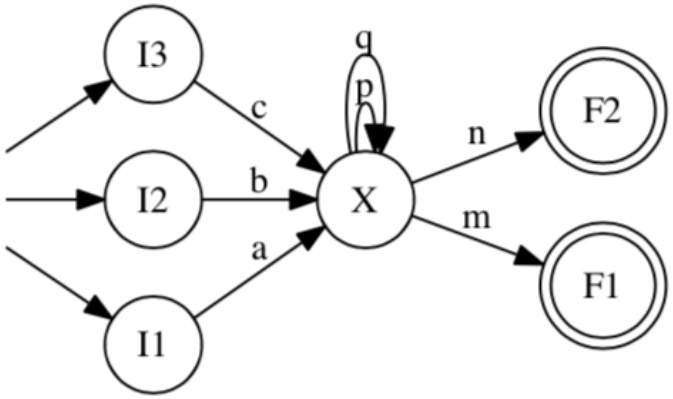
Let's convert this to an RE and check our work against this RE

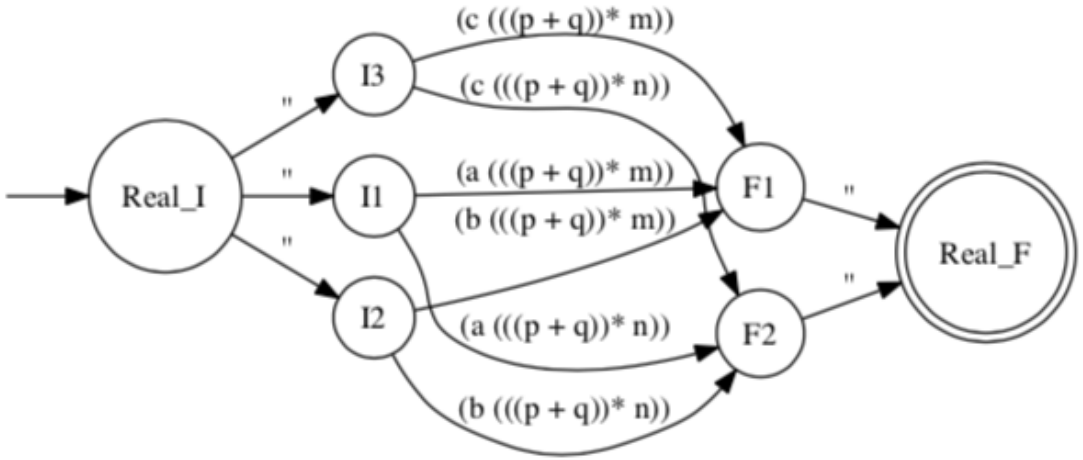
NFA to RE conversion

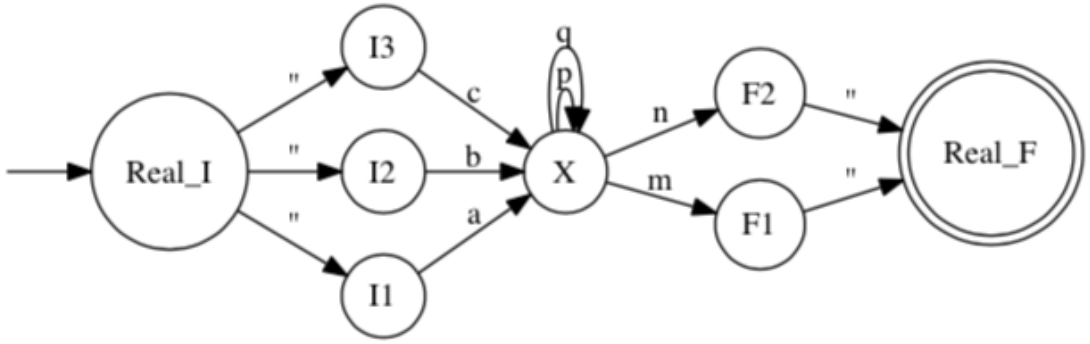


(space for work)

NFA to RE conversion: Results

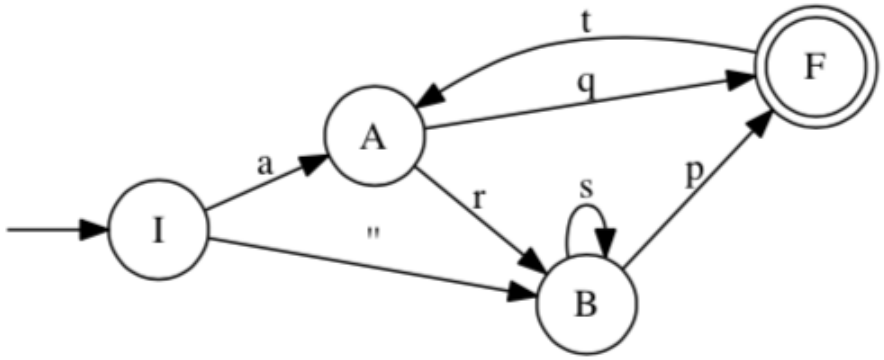






Another example of NFA to RE

(space for work)



One more example of RE -> NFA -> DFA

ac* + acd*, simplified as a(c*+cd*)

Postage-stamp problems

- For relatively prime numbers a,b what is the largest integer not expressible as a linear combination of a, b ?
  - Let a = 2, b = 3
  - (for larger numbers, let's use a tool)
- How about for more pairwise relatively prime numbers a,b,c
- How about non relatively-prime numbers?

Review

Find the strings in the language of these RE

- (00*1 + 11*01)*
- ((00*1)* + 11*01)*
- (00*1 + (11*01)*)*

```
: 1 iso_dfa(
  2     min_dfa(
  3         nfa2dfa(
  4             re2nfa( " (00*1 + 11*01)* " ))),
  5     min_dfa(
  6         nfa2dfa(
  7             re2nfa( " ( (00*1)* + 11*01)* " )))
  8 )
```

```
Generating LALR tables
Generating LALR tables
```

```
: True
```

- Find out by developing a min DFA
  - Use iso_dfa

Compare these RE pairwise

- (00*1)*
- (0 (0+1)* 1)*

Compare these RE pairwise

- (00*1 + 11*01)*
- (0 (0+1)* 1 + 11*01)*