Extensions

Handling Transducers

Transducers

```
var x = htmlEscape(name);
var y = escapeString(x);
nameElem.innerHTML = '<a onclick=' +
    '"viewPerson(\'' + y + '\')">' + x + '</a>';
```

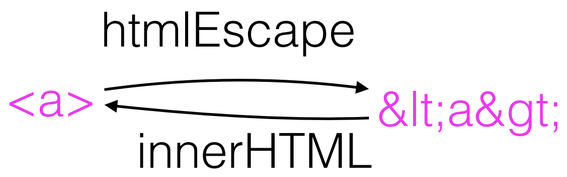
nameElem has to match

e1 = ……

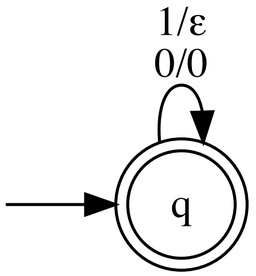
Concatenation (+)

Regular constraints (e1)

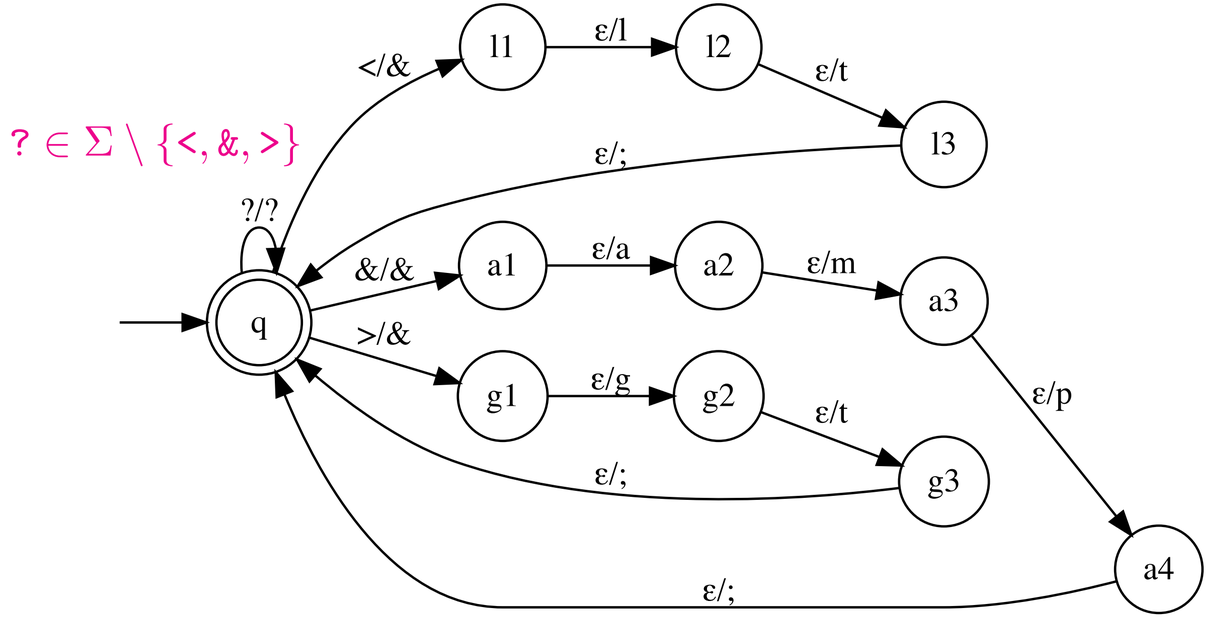
Transducers:



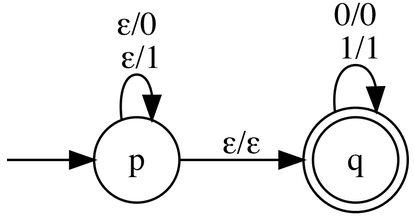
Examples of Transducers



Erase all occurrences of 1



Replace: < by <, > by >, and & by &



Input is a suffix of output

Transducer models for htmlEscape, innerHTML, ··· exist

Transducers

```
var x = htmlEscape(name);
var y = escapeString(x);
nameElem.innerHTML = '<a onclick=' +
    '"viewPerson(\'' + y + '\')">' + x + '</a>';
```

nameElem has to match

e1 = ……

x = R1(name);
y = R2(x);
z = w1 . y . w2 . x . w3;
nameElem_innerHTML := R3(z);
assert(nameElem_innerHTML matches e1)

These R1, R2, and R3 are appropriate finite transducers

Backward + Forward Propagation

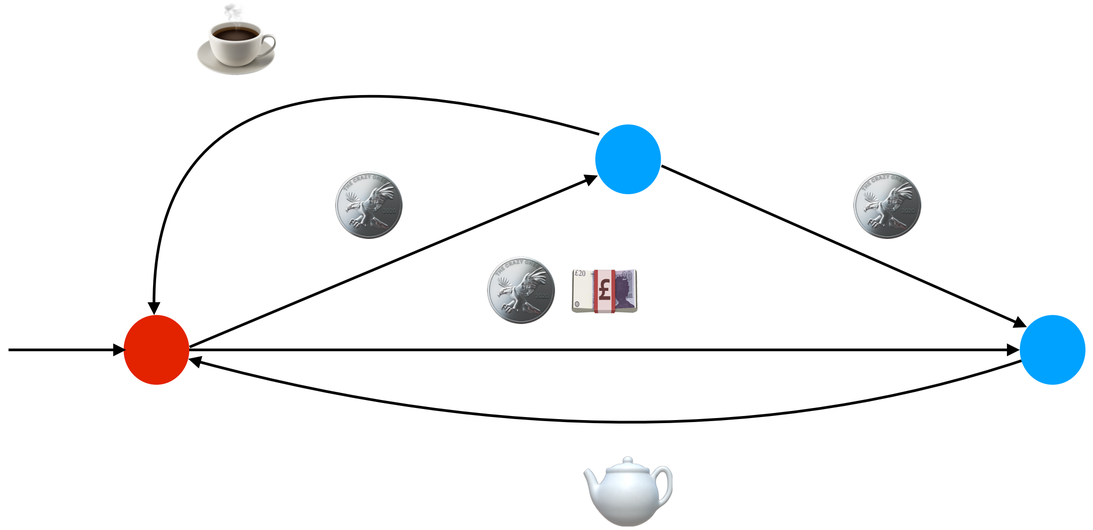
Proposition: given an NFA A and a transducer T, then both $T(L(A))$ and $T^{-1}(L(A))$ are regular and NFA for these are poly-time constructible.

We go through a proof sketch for T of the form $\text{replaceAll}_{a,\beta} : \Sigma^* \to \Sigma^*$

where $a \in \Sigma, \beta \in \Sigma^*$

Forward is not difficult …

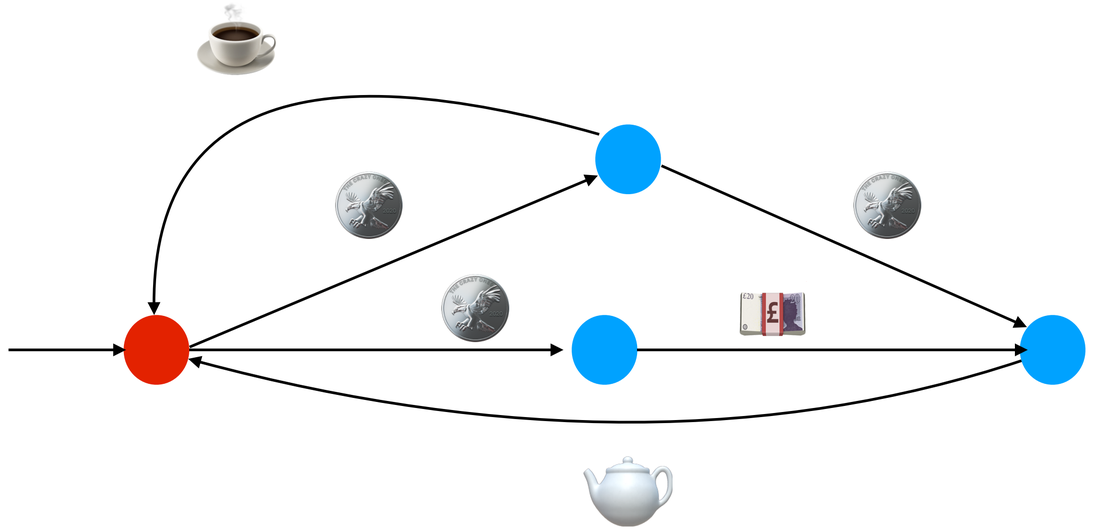
$A =$



What is replaceAll$_{\text{💷}/\text{🪙}}(L(A))$?

Forward is not difficult …

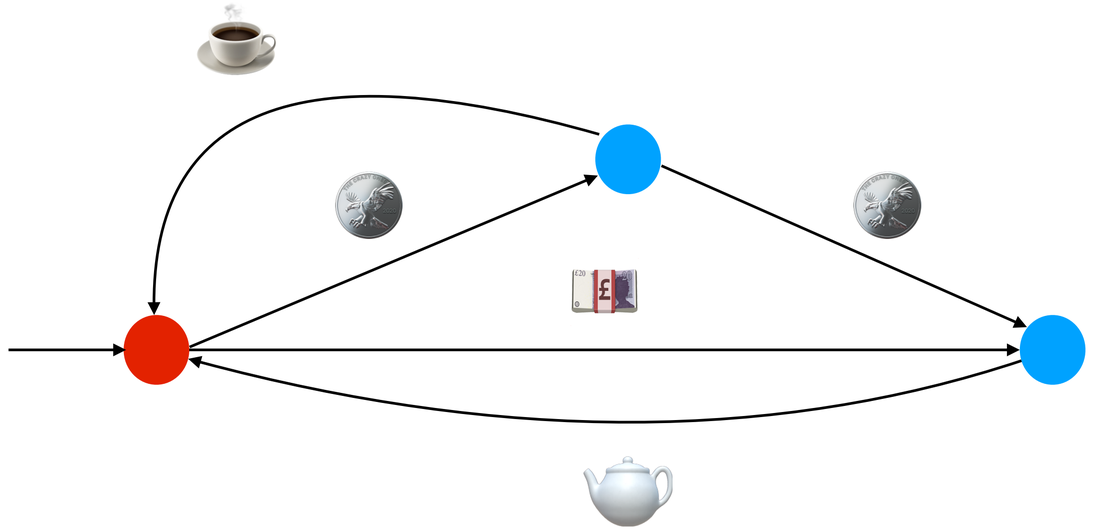
$A =$



What is replaceAll$_{\text{💷}/\text{🪙}\text{💷}}(L(A))$?

Forward is not difficult …

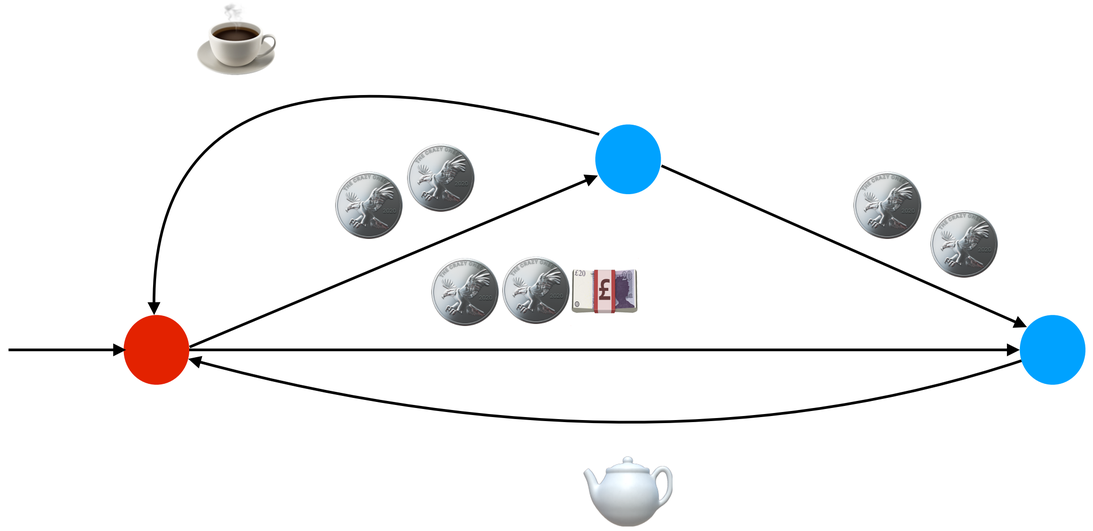
$A =$



What is replaceAll☕/🪙🪙(replaceAll💷/🪙💷($L(A)$))?

Forward is not difficult …

$A =$



What is replaceAll🪙/🪙🪙(replaceAll💷/🪙💷($L(A)$))?

Backward Propagation

$$A = (\Sigma, Q, \Delta, q_0, F)$$

$$\text{replace}^{-1}_{a,\,\beta}(L(A)) = ?$$

Let S denote the set of pairs of states (p, q) in A such that $\beta \in L(A_{p,q})$

Erase a-transitions from A

Add (p, a, q) for each $(p, q) \in S$

Lemma: the resulting automaton recognizes all $w \in \Sigma^*$ such that $\text{replace}(w, a, \beta) \in L(A)$.

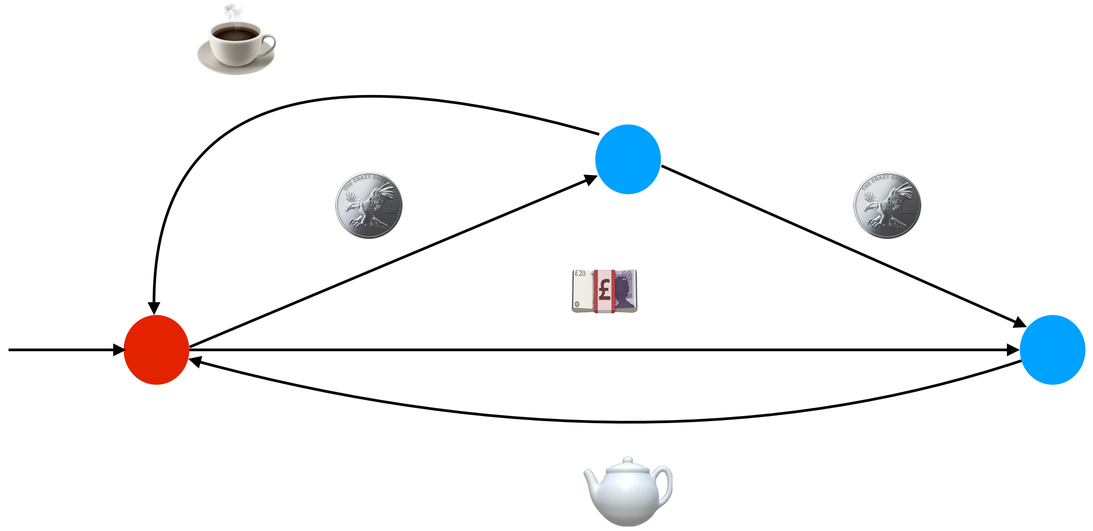
S := { pairs of states (p, q) in A such that $\beta \in L(A_{p,q})$ }

Erase a-transitions from A

Add (p, a, q) for each $(p, q) \in S$

Exercise

$A =$



What is $\text{replaceAll}^{-1}_{/\text{☕}}(L(A))$?

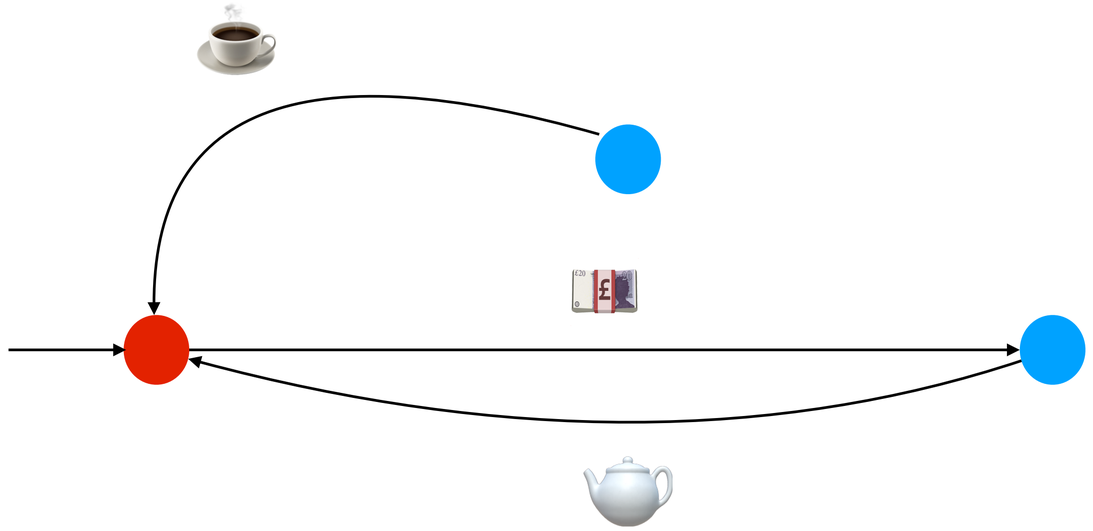
S := { pairs of states (p, q) in A such that $\beta \in L(A_{p,q})$ }

Erase a-transitions from A

Add (p, a, q) for each $(p, q) \in S$

Exercise

$A =$



What is $\text{replaceAll}^{-1}_{🪙🪙/☕}(L(A))$?

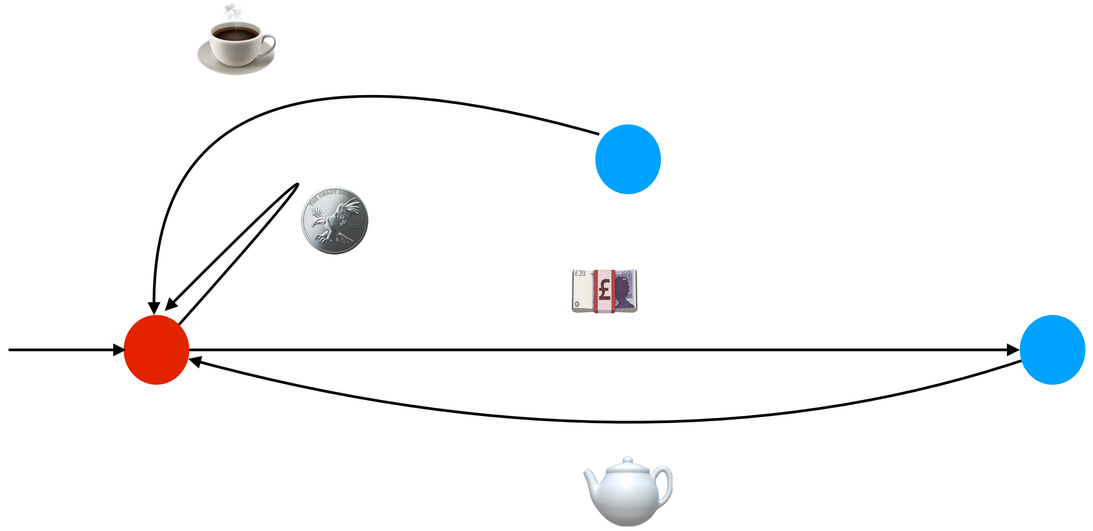
S := { pairs of states (p, q) in A such that $\beta \in L(A_{p,q})$ }

Erase a-transitions from A

Add (p, a, q) for each $(p, q) \in S$

Exercise

$A =$



What is $\text{replaceAll}^{-1}_{\text{🪙}/\text{☕}}(L(A))$?

Exercises

1. Extend forward/backward propagation to general replaceAll$_{a,\beta} : \Sigma^* \to \Sigma^*$. Think of different matching strategies (e.g. first/nondeterministic)
2. Extend forward/backward propagation to transducers
3. Extend backward propagation to replaceAll$_{a,x} : \Sigma^* \to \Sigma^*$, where x is a variable. Show this fails for forward propagation
4. (Challenging**) Extend forward/backward propagation to replace with capture groups. [Hint: use streaming string transducers.]