

# Finite State Transducers

Natalie Parde

UIC CS 421

# What are finite state transducers?

---

**Finite State Transducer (FST):** A type of FSA that describes mappings between two sets of items

---

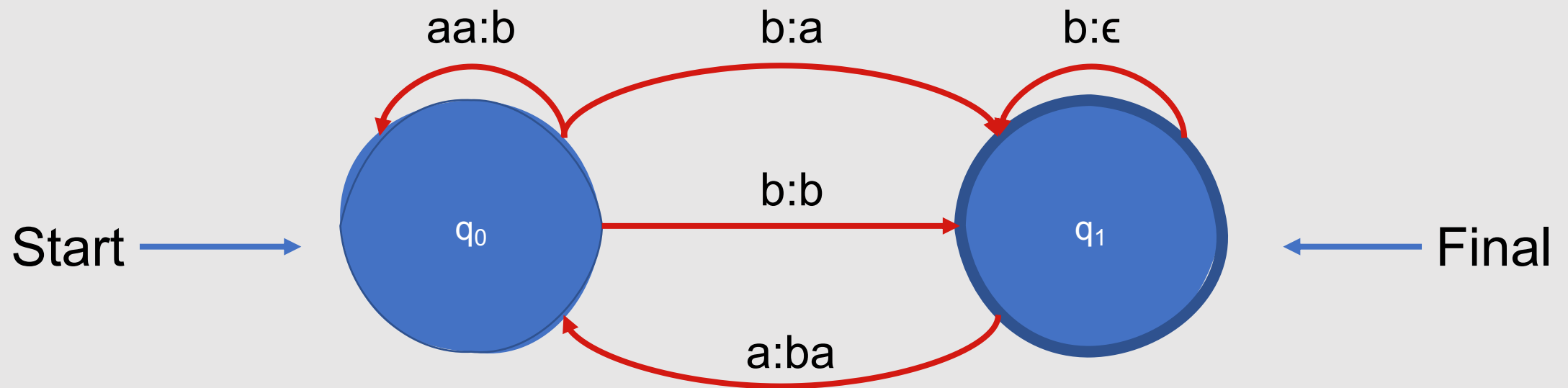
This means that FSTs recognize or generate pairs of items

---

FSAs can be converted to FSTs by labeling each arc with two items (e.g., **a:b** for an input of **a** and an output of **b**)

---

# Example: Simple FST



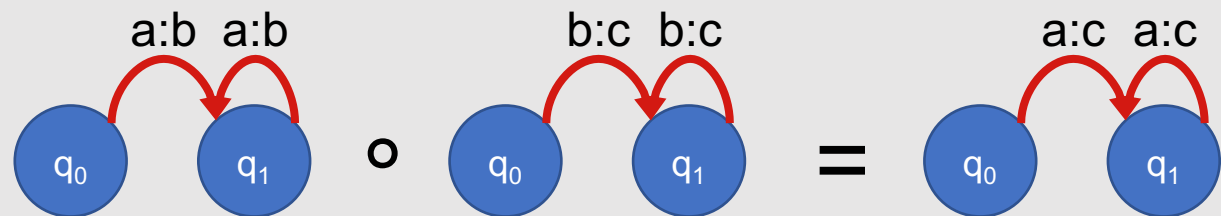
# Formal Definition

- A finite state transducer can be specified by enumerating the following properties:
  - The set of states,  $Q$
  - A finite input alphabet,  $\Sigma$
  - A finite output alphabet,  $\Delta$
  - A start state,  $q_0$
  - A set of accept/final states,  $F \subseteq Q$
  - A transition function or transition matrix between states,  $\delta(q, i)$
  - An output function giving the set of possible outputs for each state and input,  $\sigma(q, i)$
- $\delta(q, i)$ : Given a state  $q \in Q$  and input  $i \in \Sigma$ ,  $\delta(q, i)$  returns a new state  $q' \in Q$ .

# Formal Properties

**Composition:** Letting  $T_1$  be an FST from  $I_1$  to  $O_1$  and letting  $T_2$  be an FST from  $I_2$  to  $O_2$ , the two FSTs can be composed such that the resulting FST maps directly from  $I_1$  to  $O_2$ .

**Inversion:** Letting  $T$  be an FST that maps from  $I$  to  $O$ , its inversion ( $T^{-1}$ ) will map from  $O$  to  $I$ .



# Deterministic vs. Non- Deterministic FSTs

---

Just like FSAs, **FSTs can be non-deterministic** ...one input can be translated to many possible outputs!

---

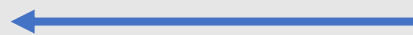
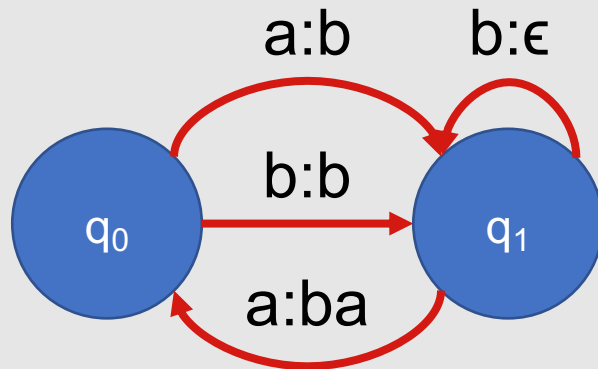
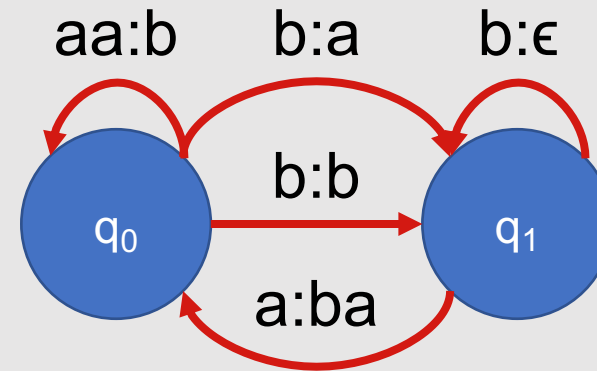
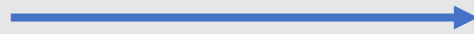
Unlike FSAs, **not all non-deterministic FSTs can be converted to deterministic FSTs**

---

FSTs with underlying deterministic FSAs (at any state, a given input maps to at most one transition out of the state) are called **sequential transducers**

# Examples: Non-Deterministic and Sequential Transducers

Non-Deterministic



Sequential

# Morphology

- **Morphemes:**
  - Small meaningful units that make up words
  - **Stems:** The core meaning-bearing units
  - **Affixes:** Bits and pieces that adhere to stems and add additional information
    - -ed
    - -ing
    - -s
- Morphological parsing is a classic use case for FSTs



# Morphological Parsing

- The task of recognizing the component morphemes of words (e.g., foxes → fox + es) and building structured representations of those components

# Why is morphological parsing necessary?

## Morphemes can be **productive**

- Example: -ing attaches to almost every verb, including brand new words
  - “Why are you Instagramming that?”

## Some languages are very **morphologically complex**

- Uygarlastiramadiklarimizdanmissinizcasina
  - Uygar ‘civilized’ + las ‘become’
  - + tir ‘cause’ + ama ‘not able’
  - + dik ‘past’ + lar ‘plural’
  - + imiz ‘p1pl’ + dan ‘abl’
  - + mis ‘past’ + siniz ‘2pl’ + casina ‘as if’

# Finite State Morphological Parsing

Goal: Take input surface realizations and produce morphological parses as output

Surface Text	Morphological Parse
cats	cat +N +PL
cat	cat +N +SG
cities	city +N +PL
geese	goose +N +PL
goose	goose +N +SG
merging	merge +V +PresPart
caught	catch +V +Past



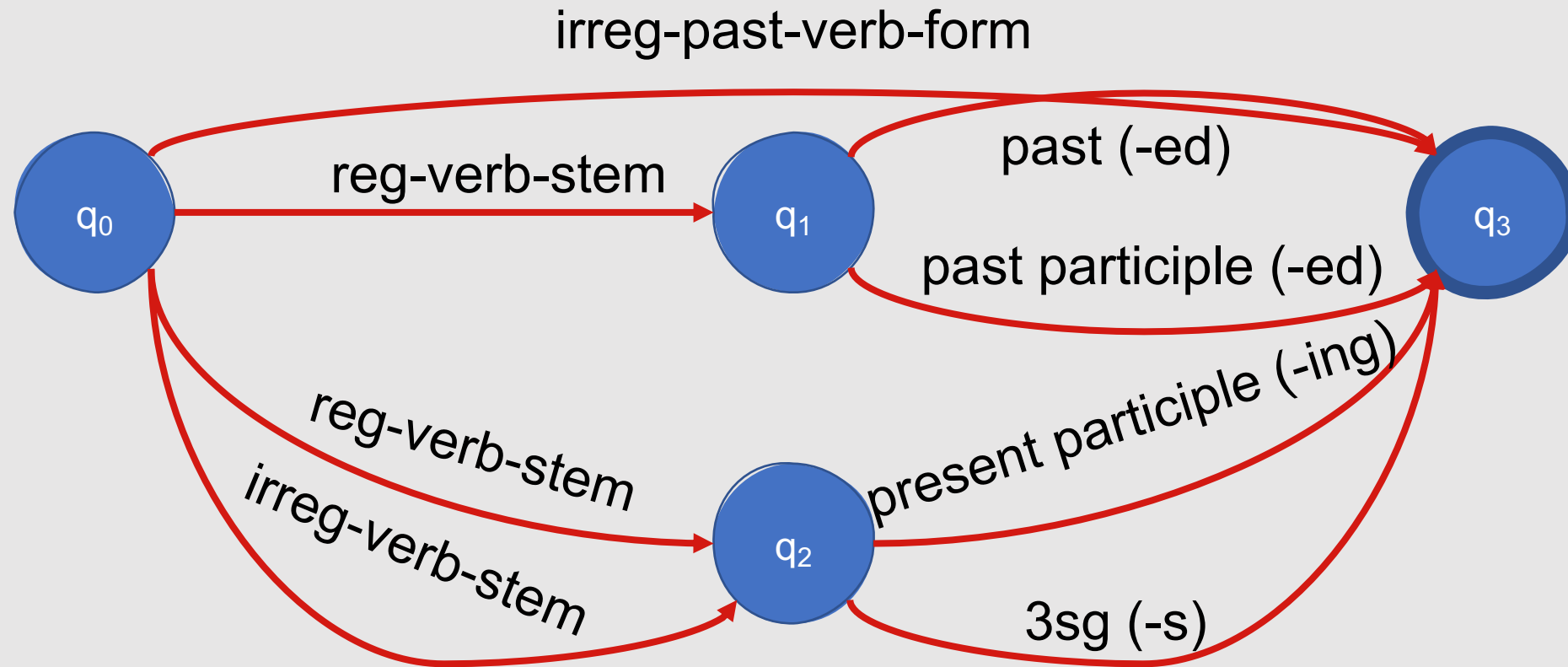
cats



cat +N +PL

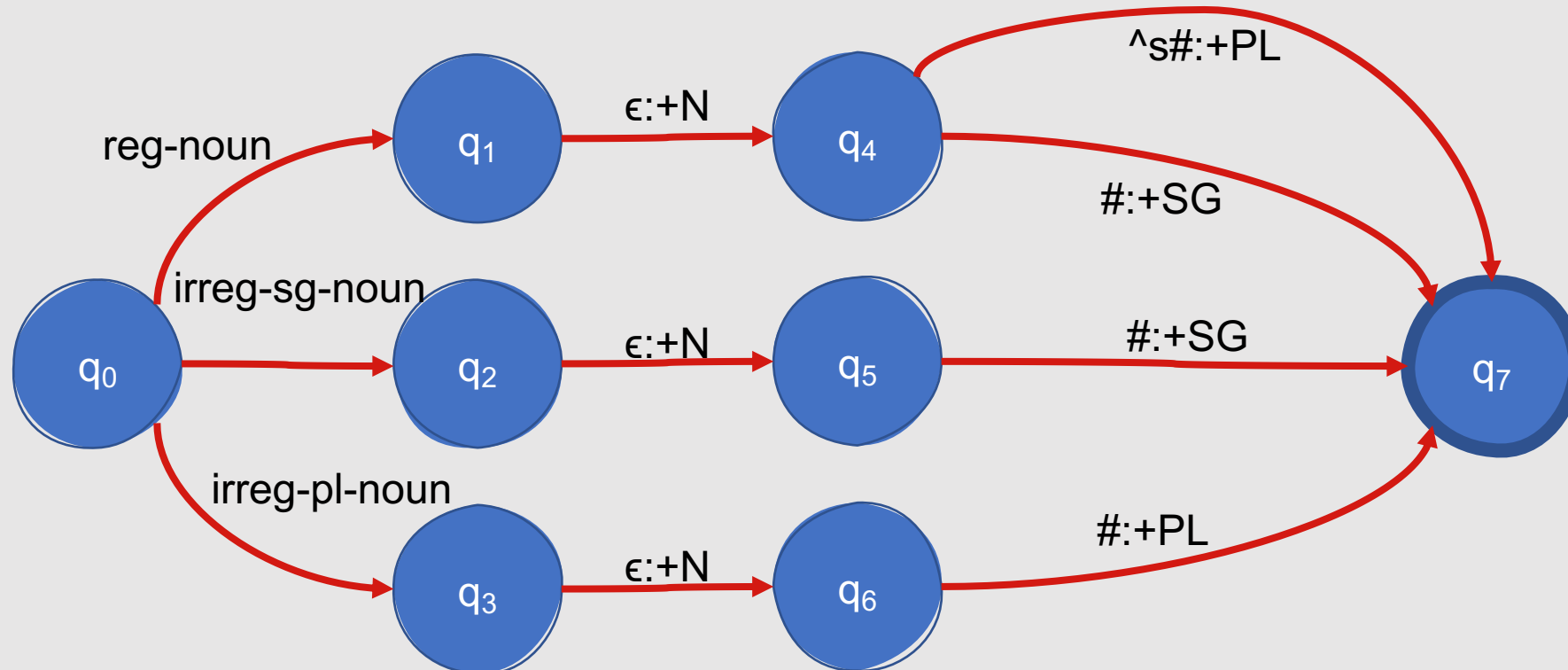


# Example Morphological Lexicon



# Finite State Morphological Parsing

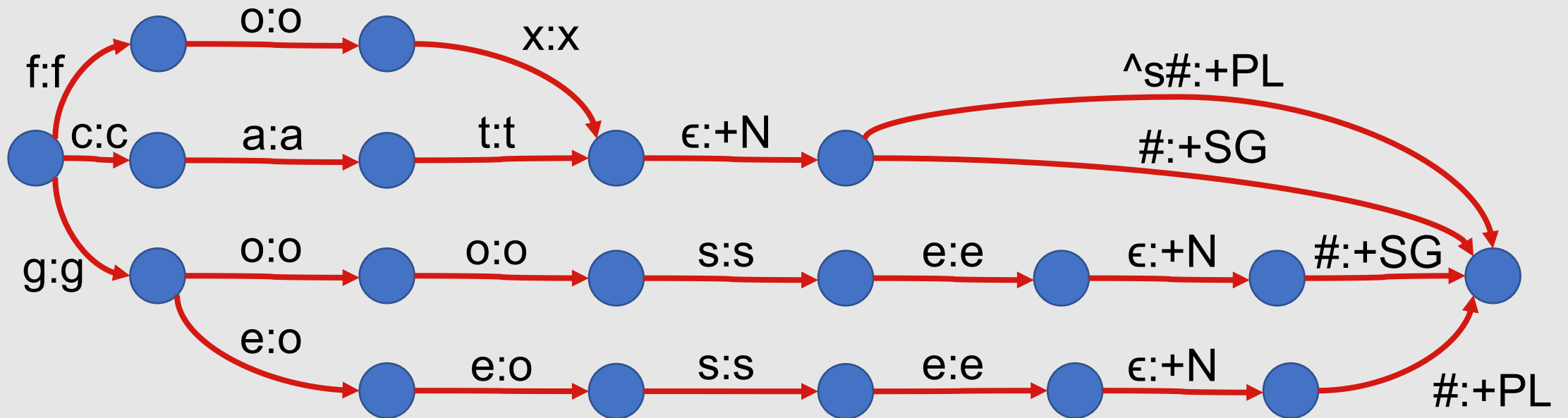
reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat		



# Finite State Morphological Parsing

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat		

foxes 🦊

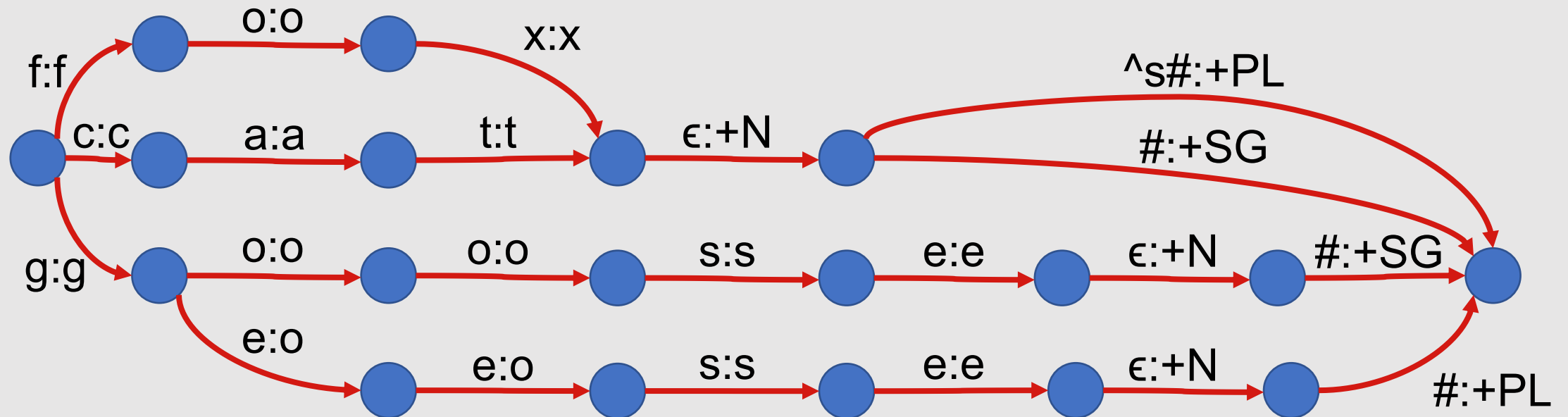


# Finite State Morphological Parsing

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat		

foxes 🦊

f

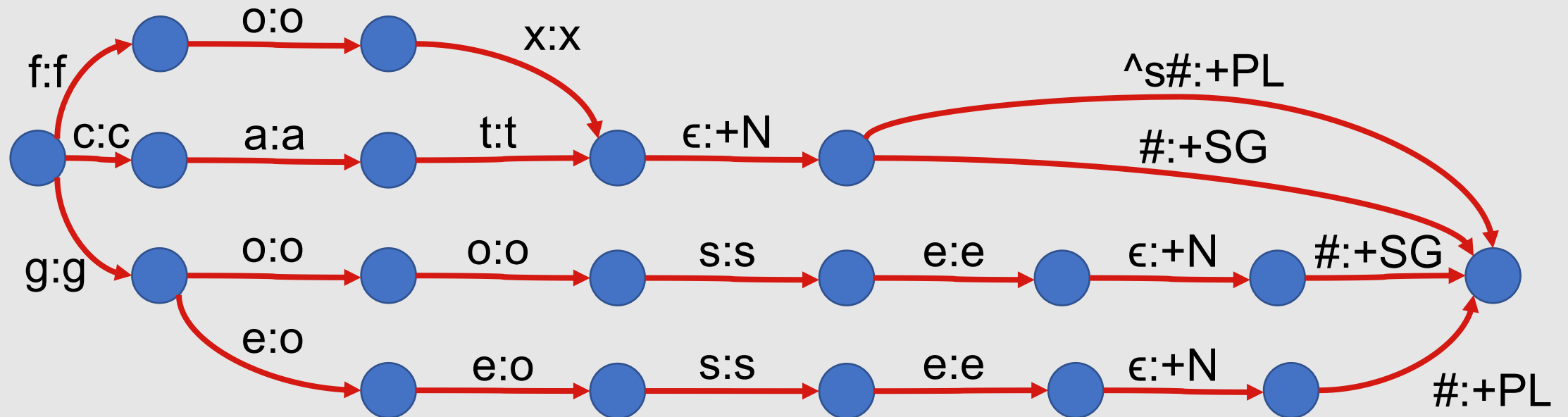


# Finite State Morphological Parsing

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat		

foxes 🦊

fo



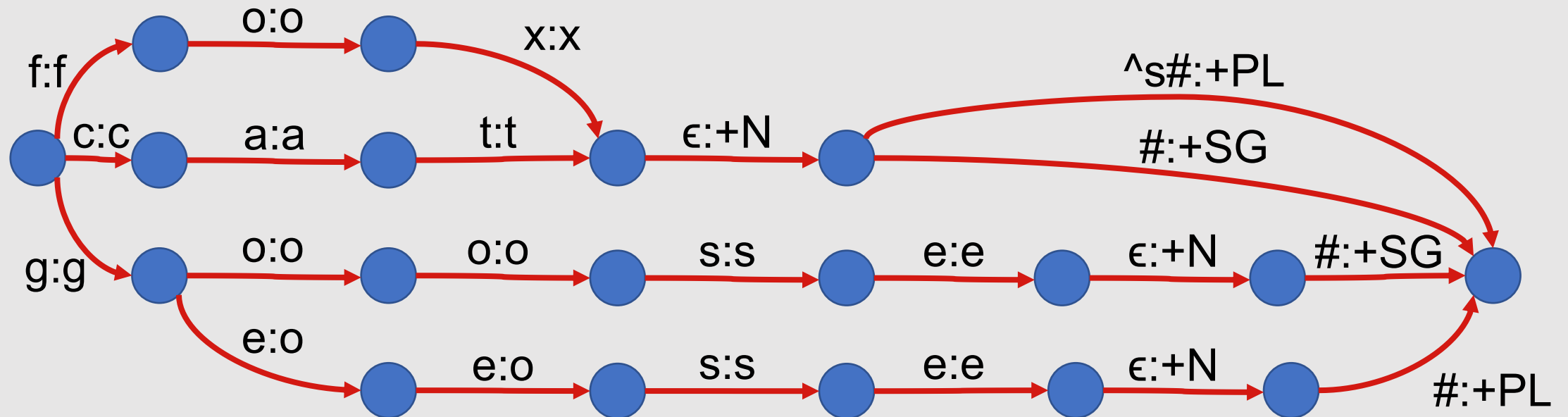


# Finite State Morphological Parsing

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat		

foxes 🦊

fox

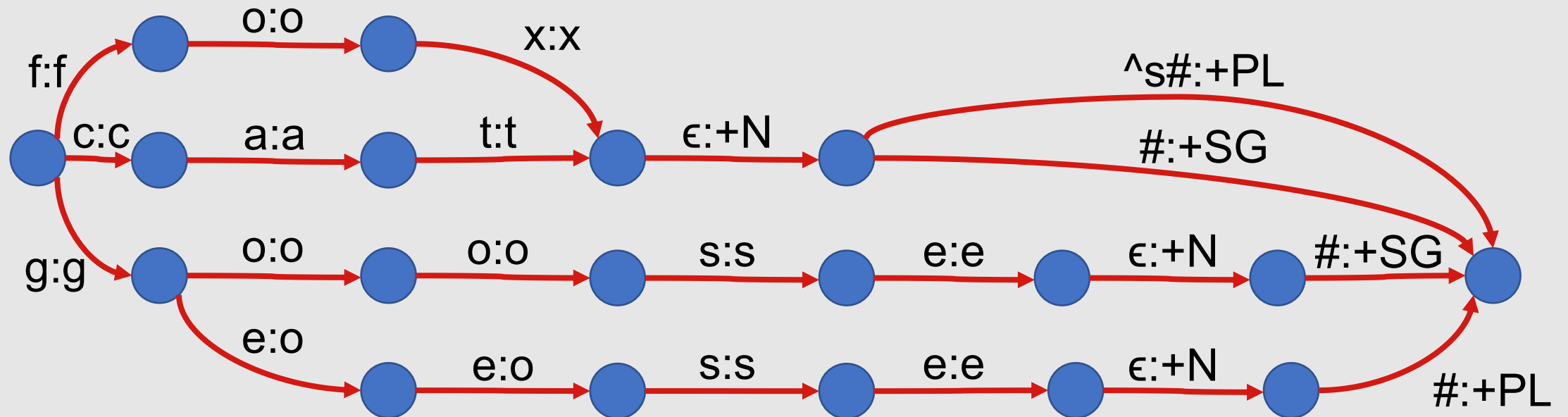


# Finite State Morphological Parsing

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat		

foxes 🦊

fox +N

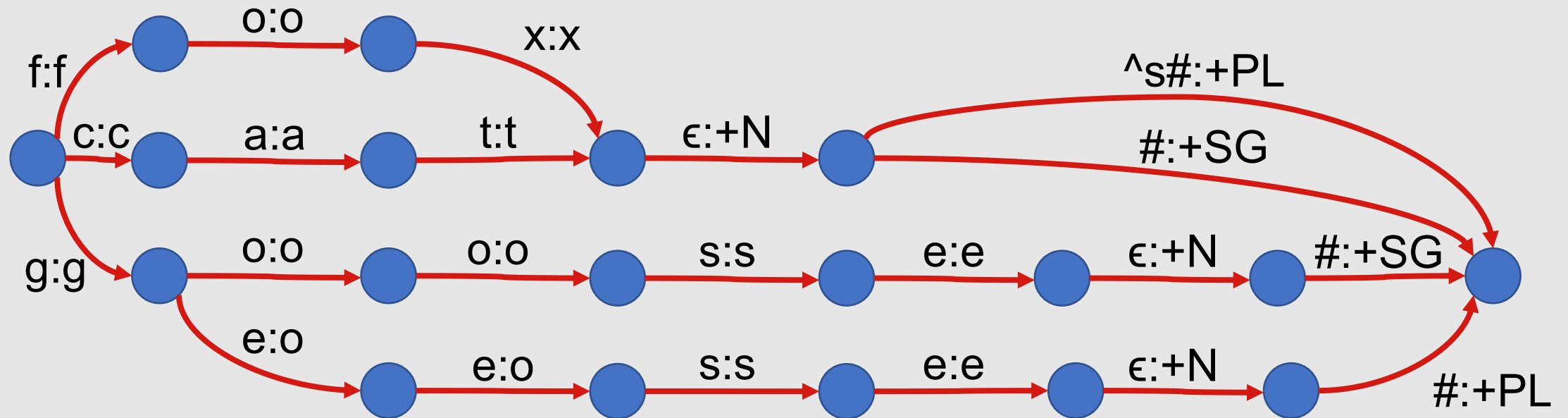


# Finite State Morphological Parsing

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat		

foxes 

fox +N

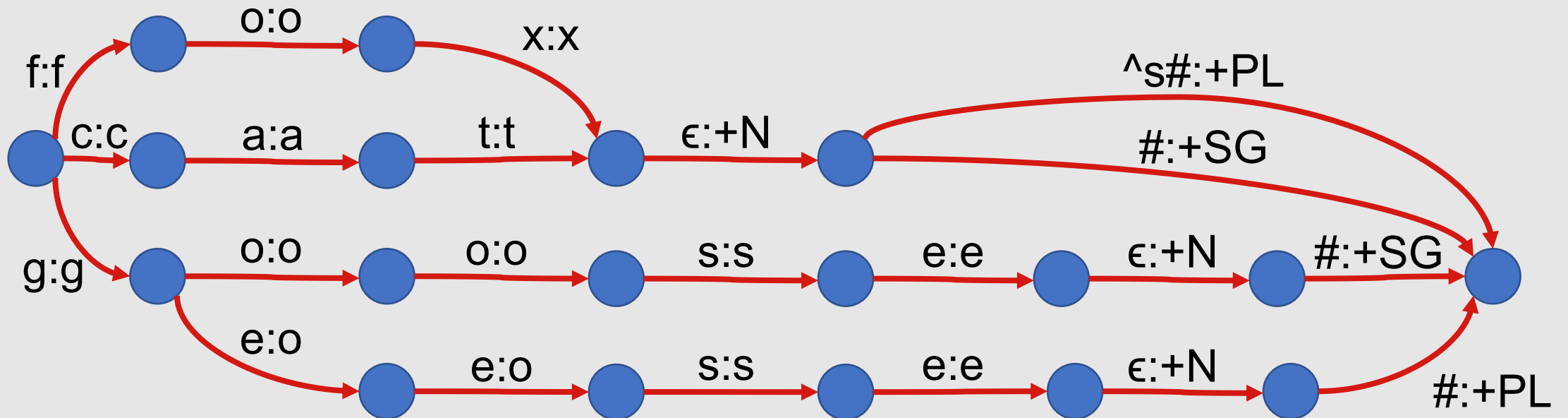


# Finite State Morphological Parsing

reg-noun	irreg-pl-noun	irreg-sg-noun
fox	g o:e o:e s e	goose
cat		

foxes 🦊

fox +N +PL



# Summary: Finite State Transducers

- FSTs are FSAs that describe mappings between two sets
- Although all non-deterministic FSAs can be converted to deterministic versions, all non-deterministic FSTs cannot
- FSTs with underlying deterministic FSAs are called sequential transducers
- FSTs are particularly useful for morphological parsing