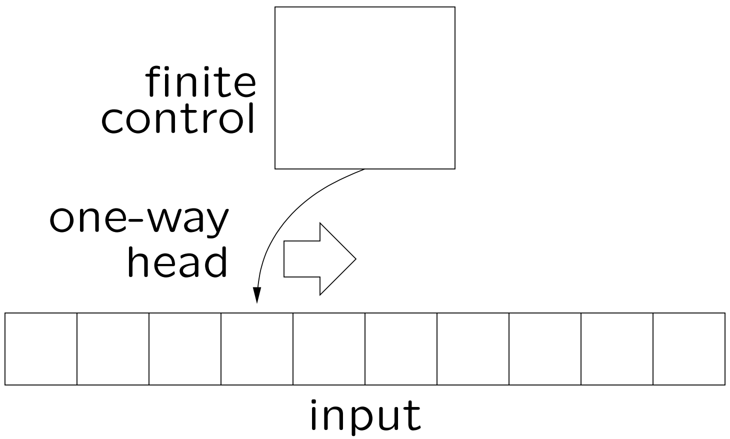
removing bidirectionality from nondeterministic finite automata

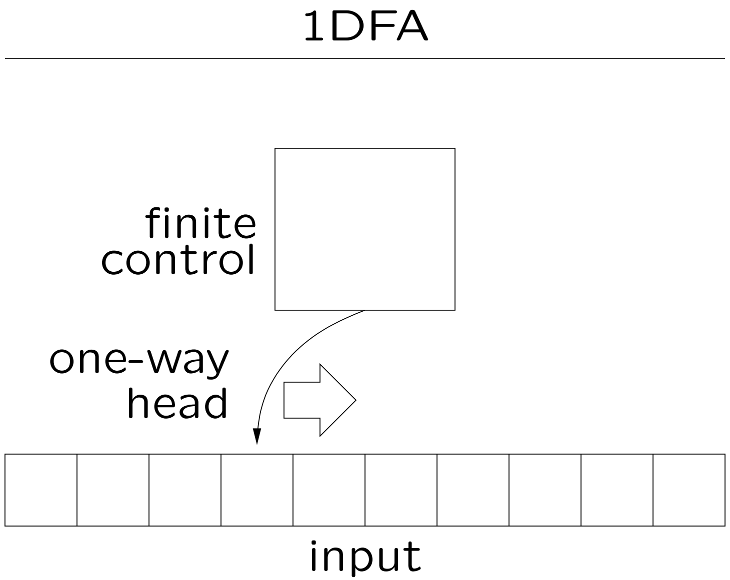
Christos Kapoutsis

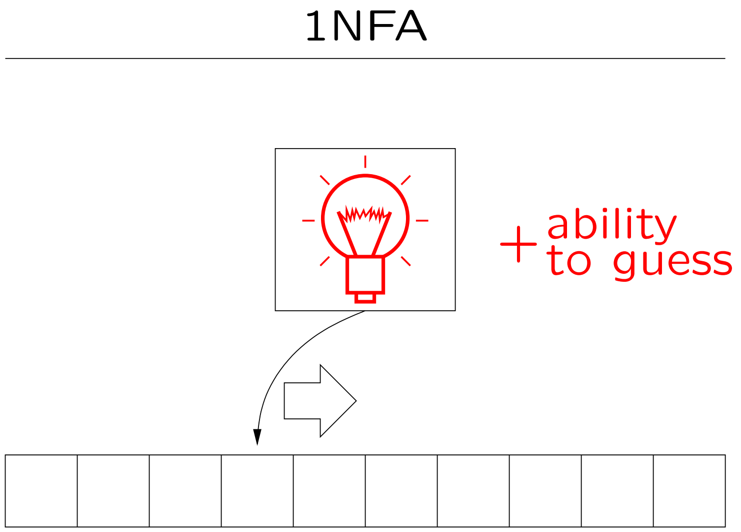
symposium on
Mathematical Foundations of Computer Science
Gdańsk, Poland, August 2005

1DFA

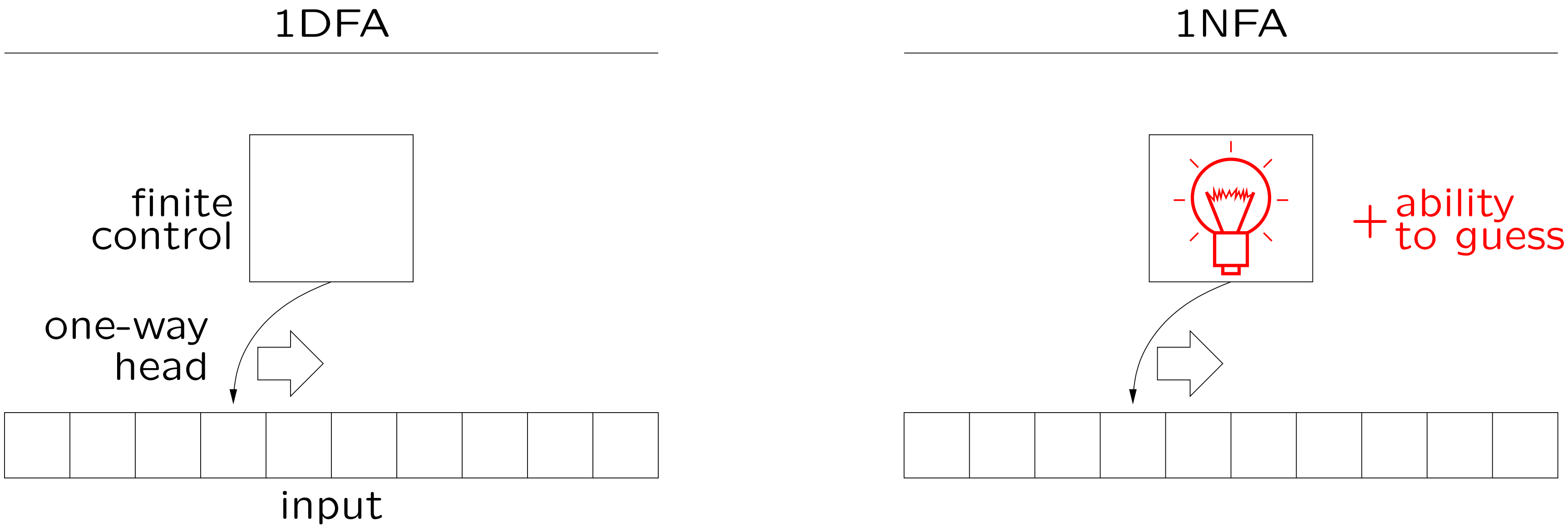


1DFAs ⟵ 1NFAs


1DFA
finite control
one-way head
input

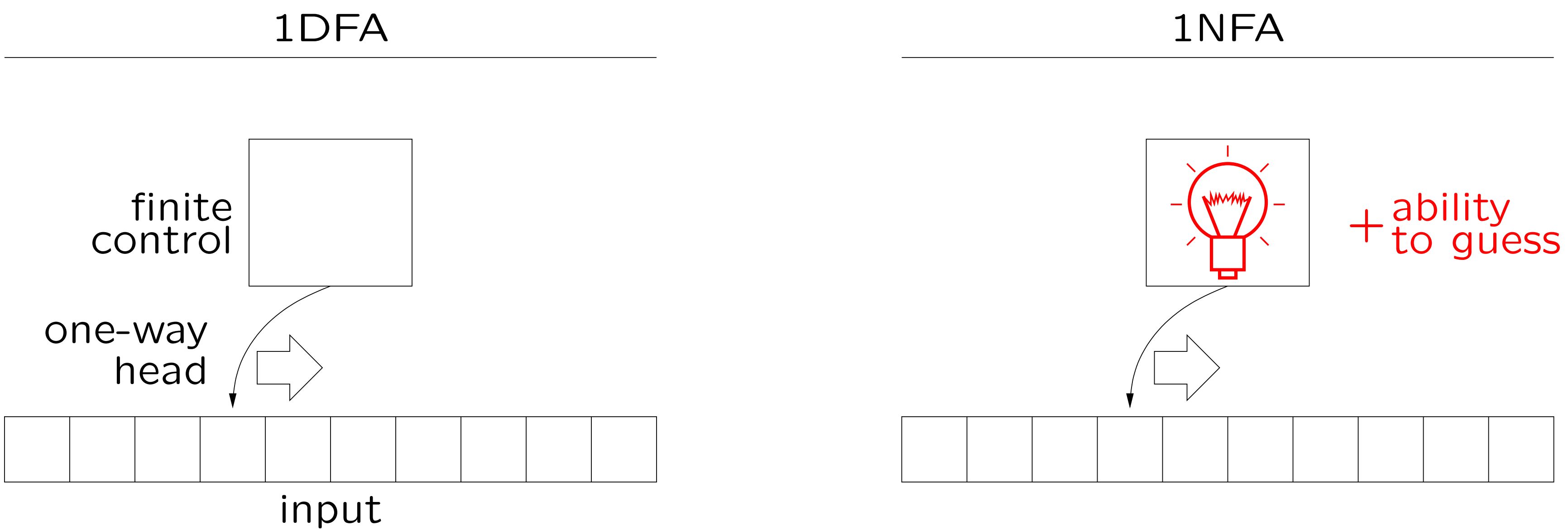

1NFA
+ ability to guess

1DFAs ⟵ 1NFAs



a 1DFA with $\leq 2^n - 1$ states ⟵ can be converted to ⟵ every 1NFA with n states

1DFAs $\longleftarrow$ 1NFAs

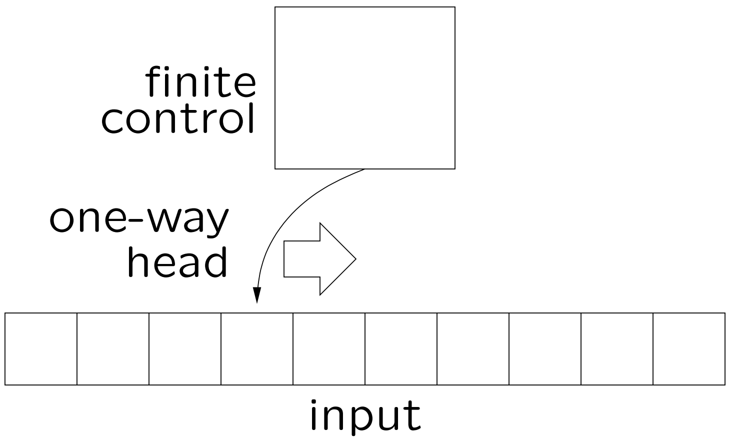


a 1DFA with $\leq 2^n - 1$ states $\xleftarrow{\text{can be converted to}}$ every 1NFA with n states

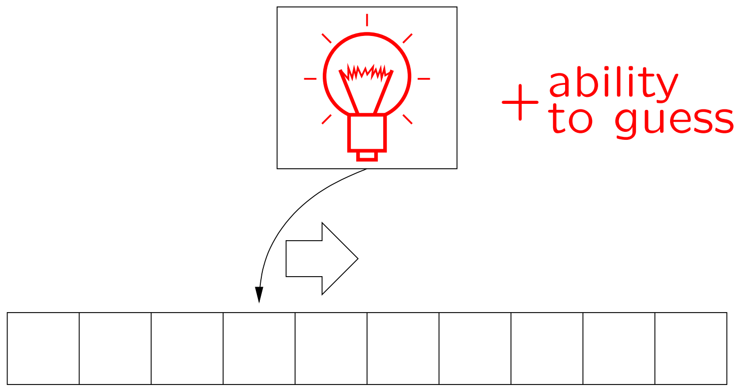
and sometimes all these $2^n - 1$ states are necessary

1DFAs $\longleftarrow$ 1NFAs

1DFA



1NFA

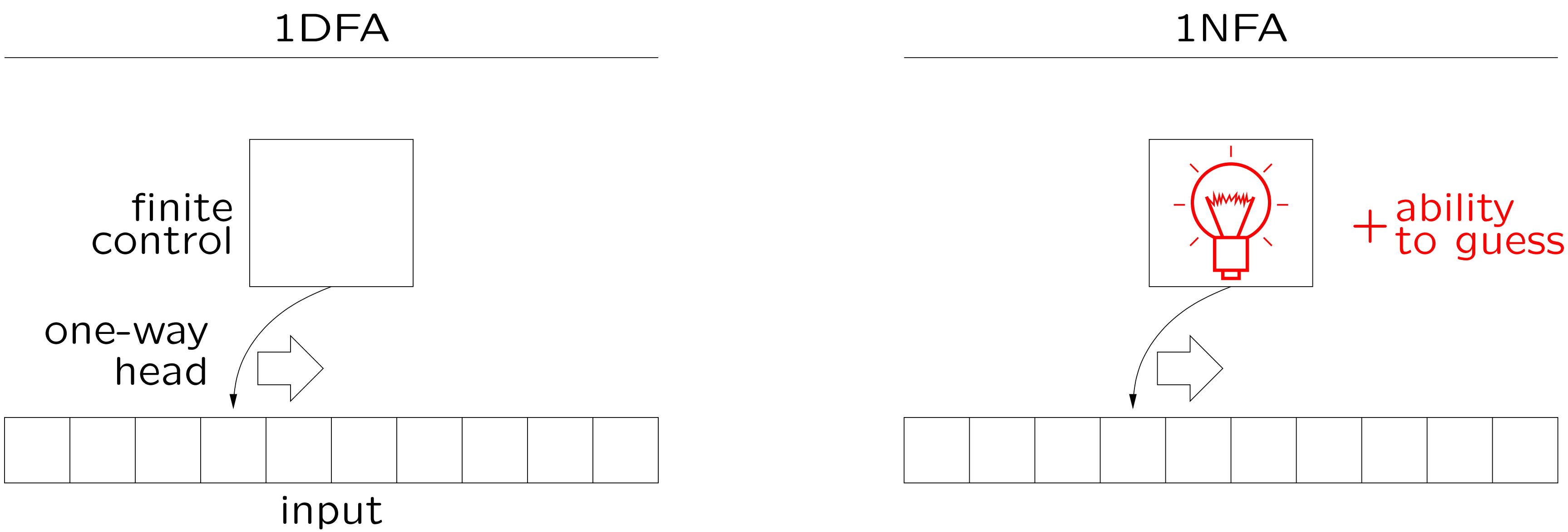


a 1DFA with $\leq 2^n - 1$ states $\longleftarrow$ (can be converted to) every 1NFA with n states

and sometimes all these $2^n - 1$ states are necessary

"the trade-off is exactly $2^n - 1$*"*

1DFAs $\longleftarrow$ 1NFAs

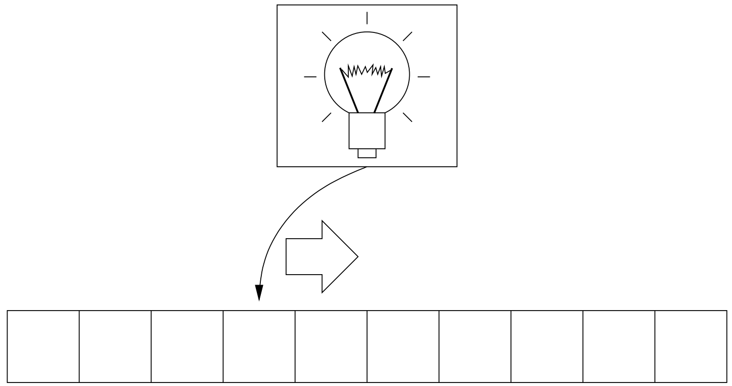


a 1DFA with $\leq 2^n - 1$ states $\xleftarrow{\text{can be converted to}}$ every 1NFA with n states

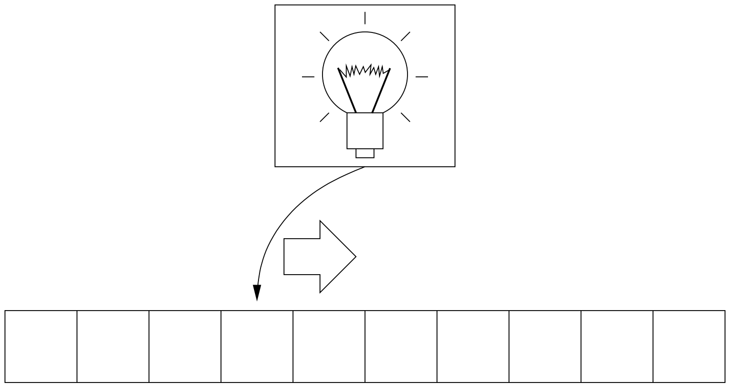
and sometimes all these $2^n - 1$ states are necessary

"SUBSET CONSTRUCTION"

1NFA

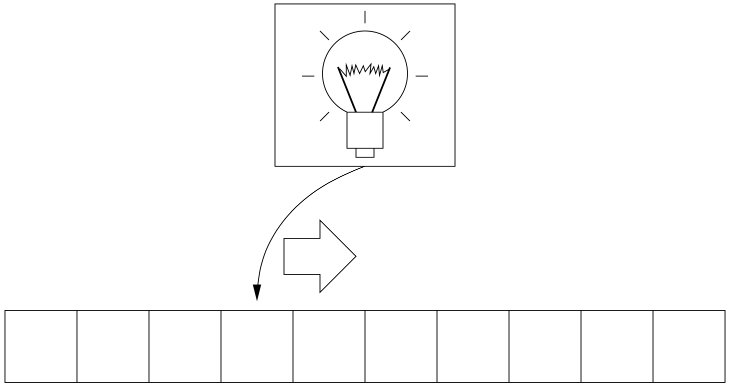


1NFA

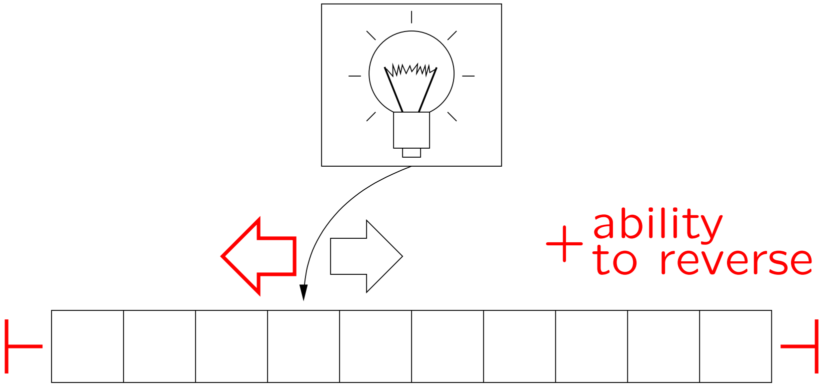


1NFAs ⟵ 2NFAs

1NFA

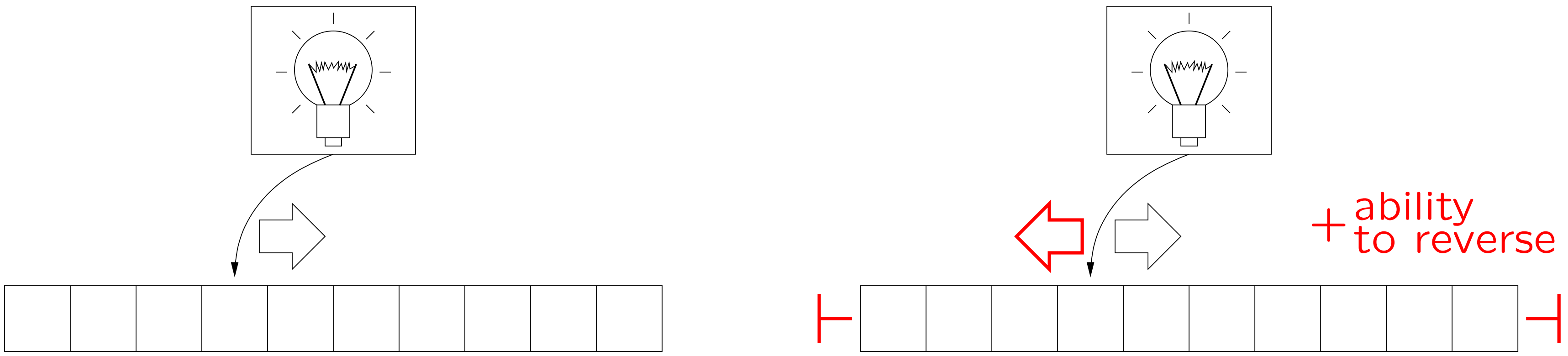


2NFA



1NFAs ⟵ 2NFAs

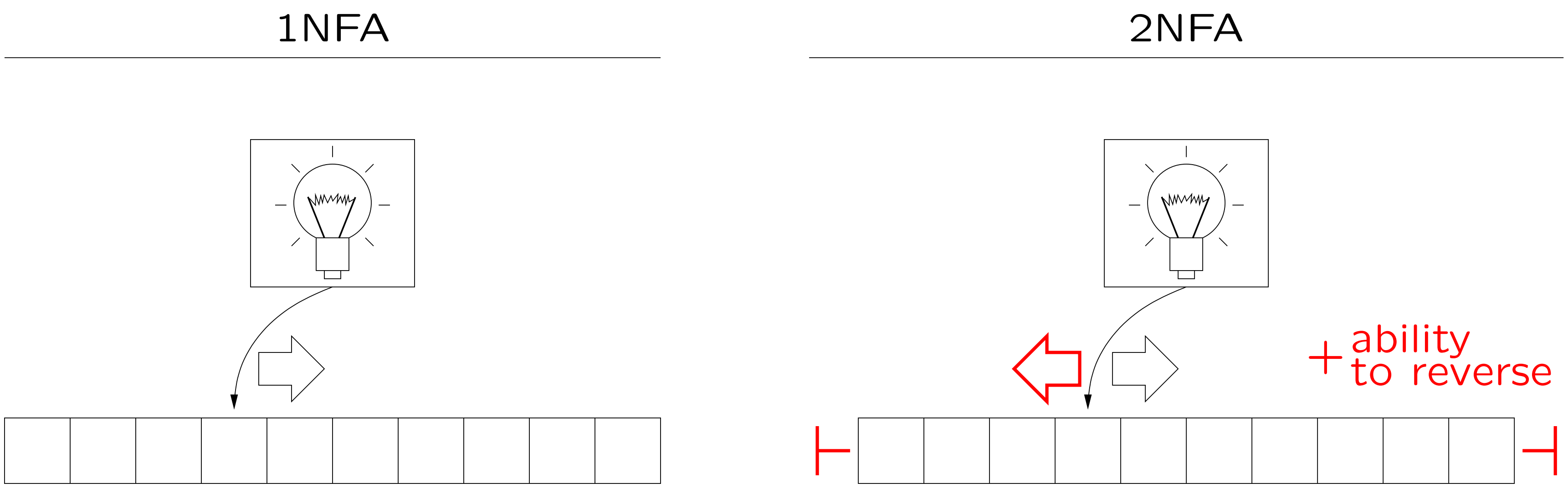
1NFA	2NFA



a 1NFA with $\leq$ [?] states ⟵ can be converted to ⟵ every 2NFA with n states

and sometimes all these [?] states are necessary

"the trade-off is exactly [?]*"*



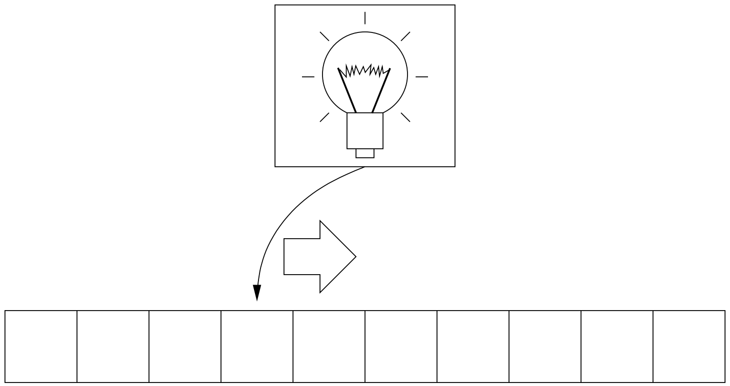
a 1NFA with $\leq$ [?] states ⟵ can be converted to ⟵ every 2NFA with n states

and sometimes all these [?] states are necessary

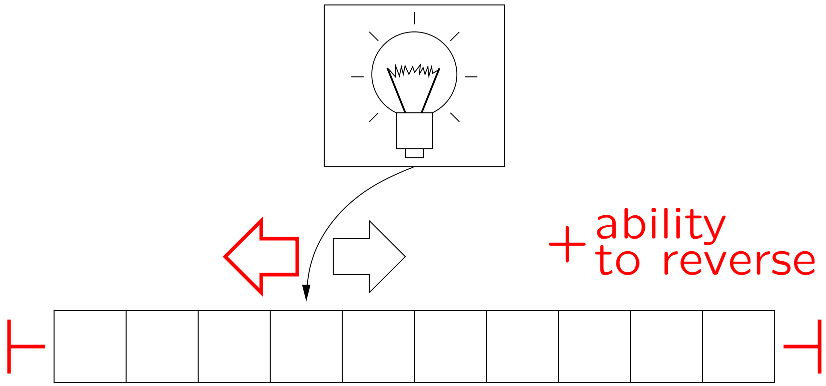
“??? CONSTRUCTION”

1NFAs $\longleftarrow$ 2NFAs

1NFA



2NFA

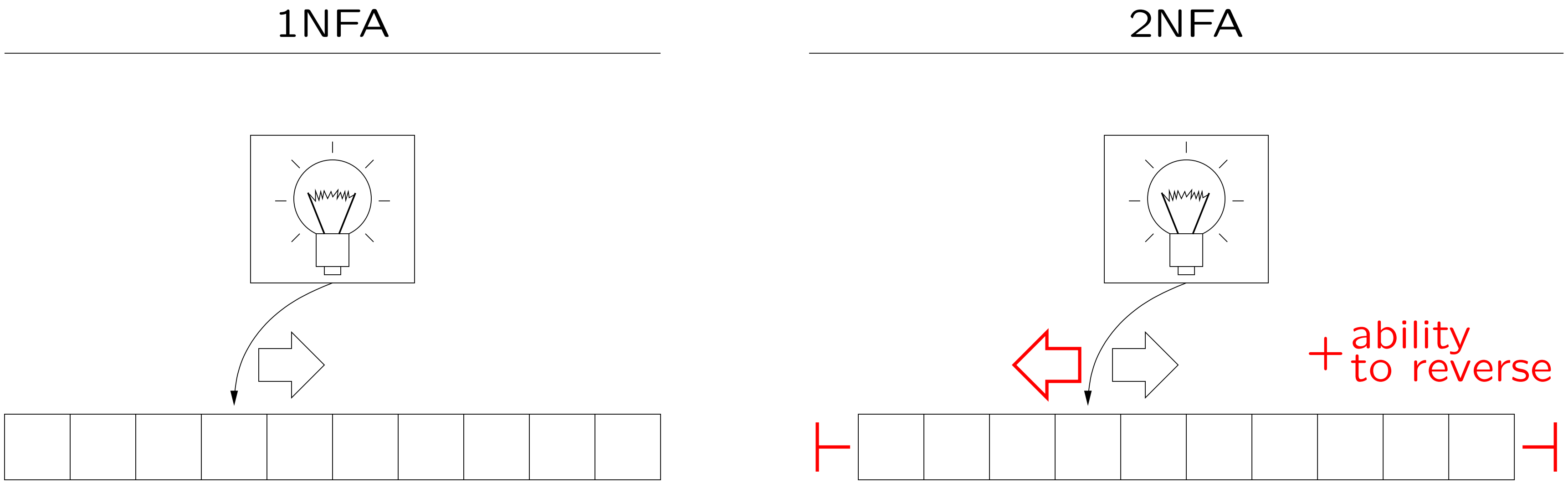


a 1NFA with $\leq \binom{2n}{n+1}$ states $\longleftarrow$ (can be converted to) every 2NFA with n states

and sometimes all these $\binom{2n}{n+1}$ states are necessary

"*the trade-off is exactly* $\binom{2n}{n+1}$"

1NFAs $\longleftarrow$ 2NFAs



a 1NFA with $\leq \binom{2n}{n+1}$ states $\longleftarrow$ can be converted to $\longleftarrow$ every 2NFA with n states

and sometimes all these $\binom{2n}{n+1}$ states are necessary

“FRONTIER CONSTRUCTION”

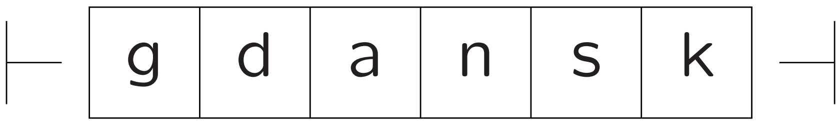
1NFAs $\longleftarrow$ 2NFAs

$$
\boxed{\;?\;} \begin{cases}
\le n2^{n^2} & \approx 2^{n^2} & \text{[Shepherdson59]} \\
\le n(n!)^2 & \approx 2^{2n \lg n} & \text{[Hopcroft-Ullman79]} \\
\le n(n+1)^n & \approx 2^{n \lg n} & \text{[think on Shepherdson]} \\
\le 2^{3n}+2 & \approx 2^{3n} & \text{[Birget93]} \\
 & & \\
= \binom{2n}{n+1} & \approx \frac{1}{\sqrt{n}} 2^{2n} & \\
 & & \\
\ge 2^{n/2} & \approx 2^{n/2} & \text{[think on Seiferas,Damanik]} \\
\ge 2^{(n-1)/2}-1 & \approx 2^{n/2} & \text{[Sakoda-Sipser78][Birget93]} \\
\ge 2^{(n-2)/4} & \approx 2^{n/4} & \text{[Seiferas73][Damanik96]}
\end{cases}
$$

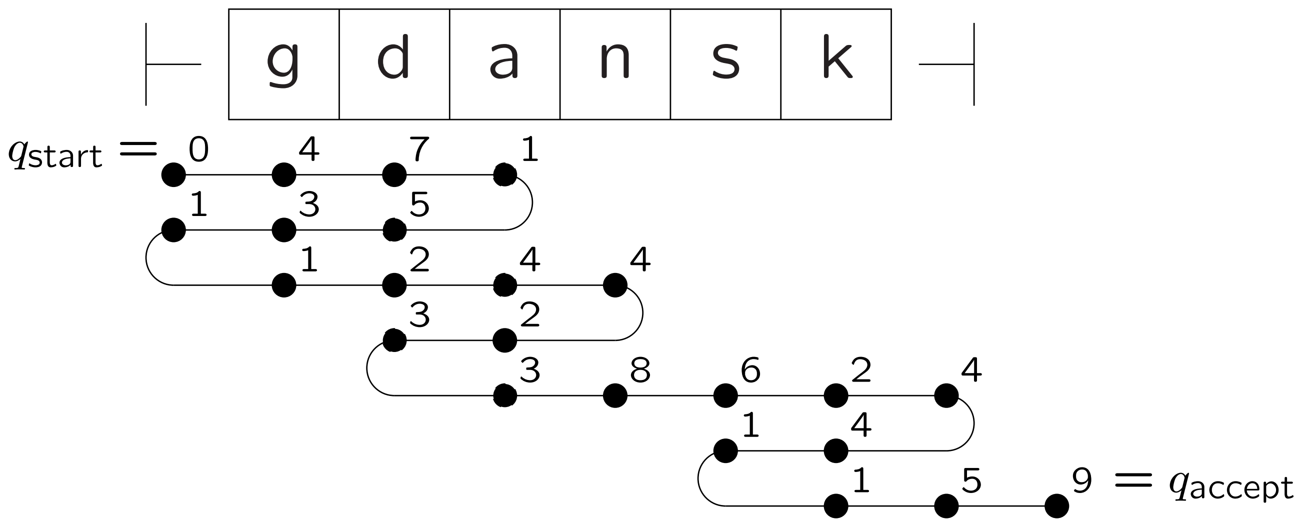
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

CROSSING-SEQUENCE CONSTRUCTION

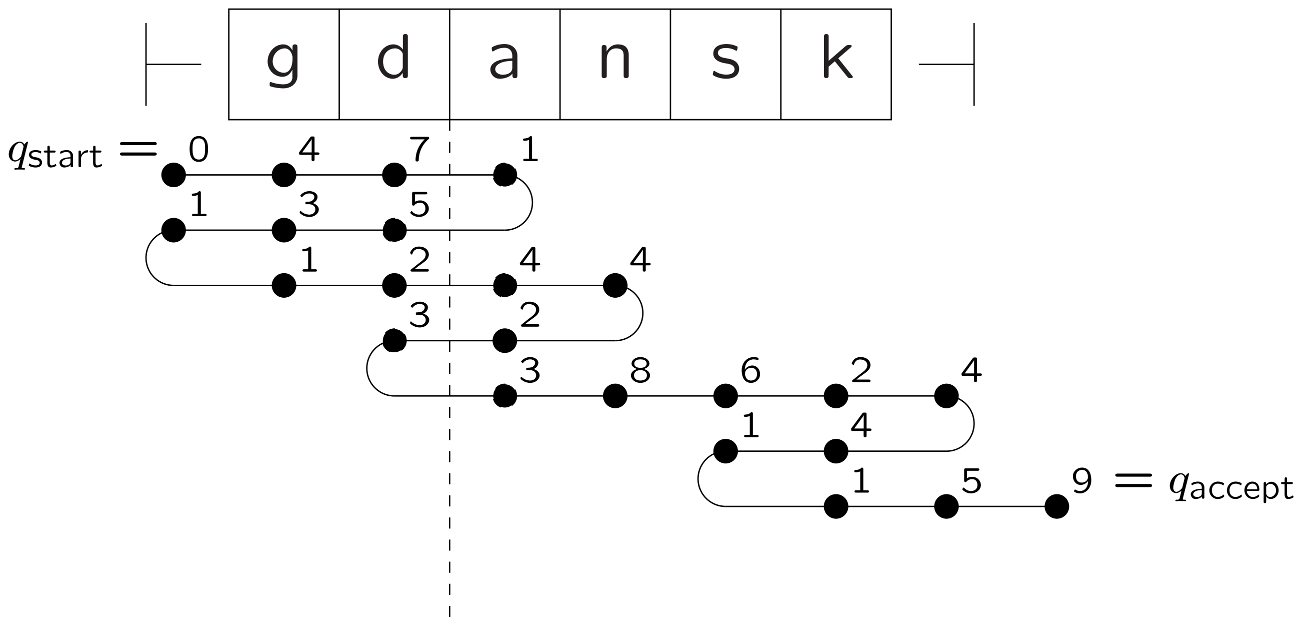
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



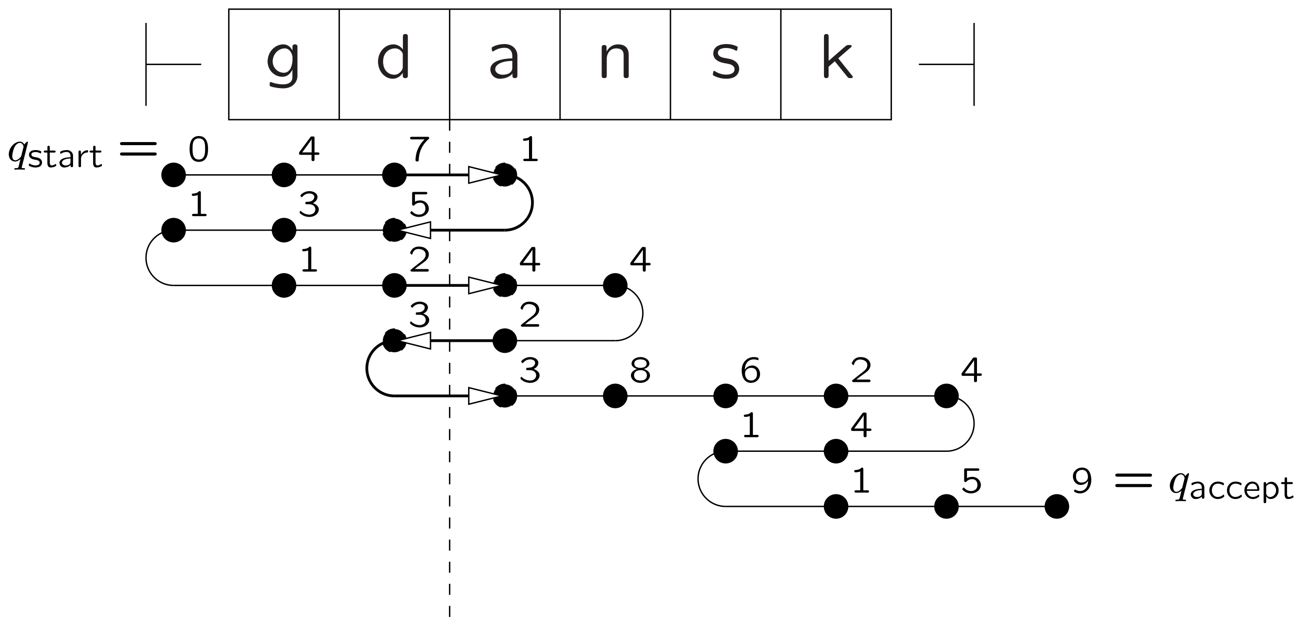
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



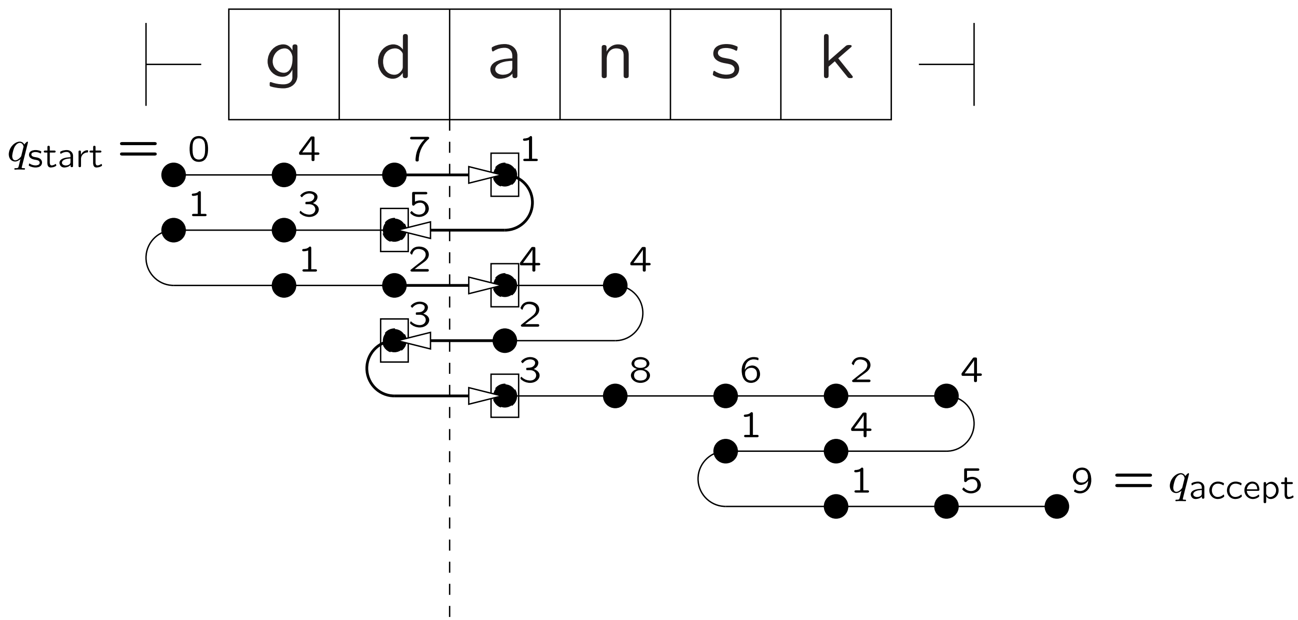
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



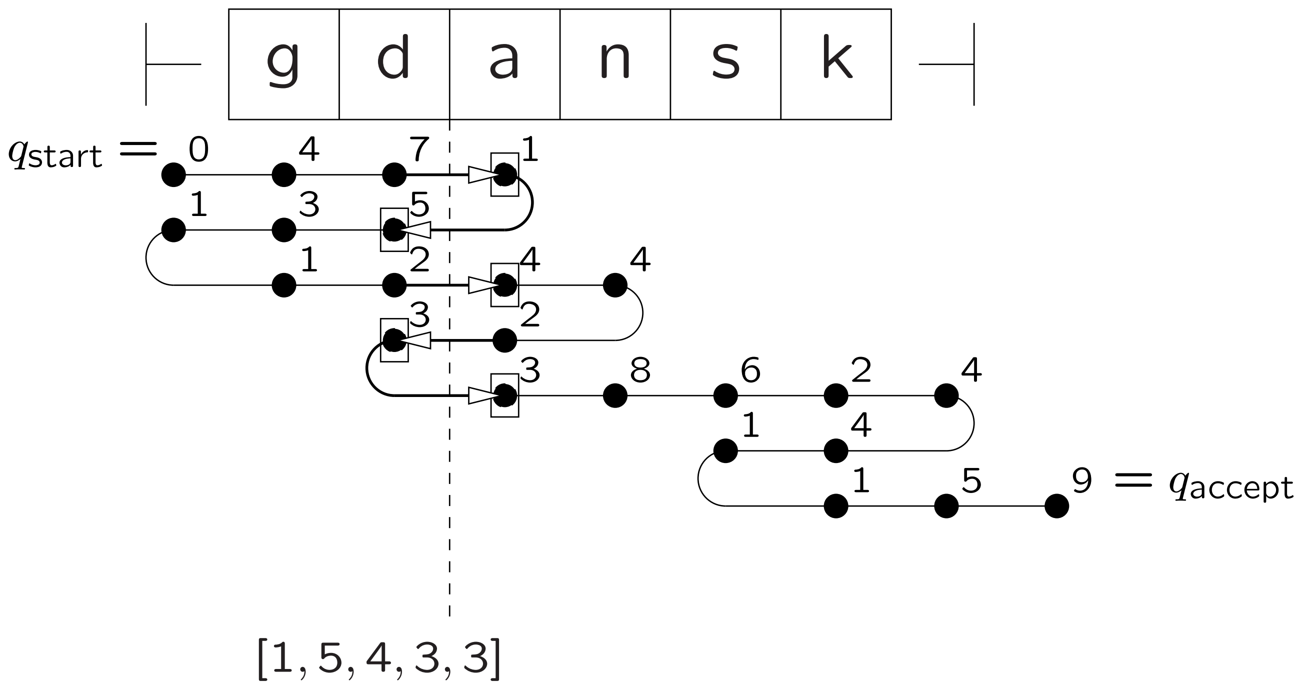
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



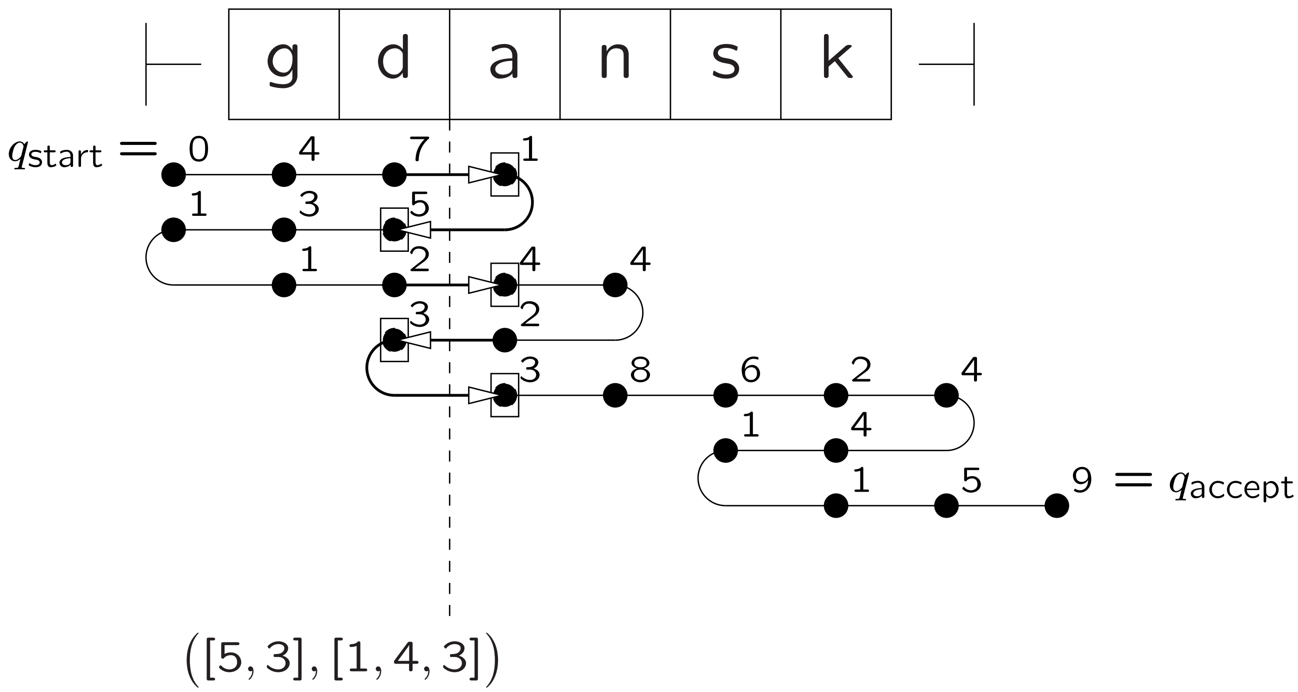
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



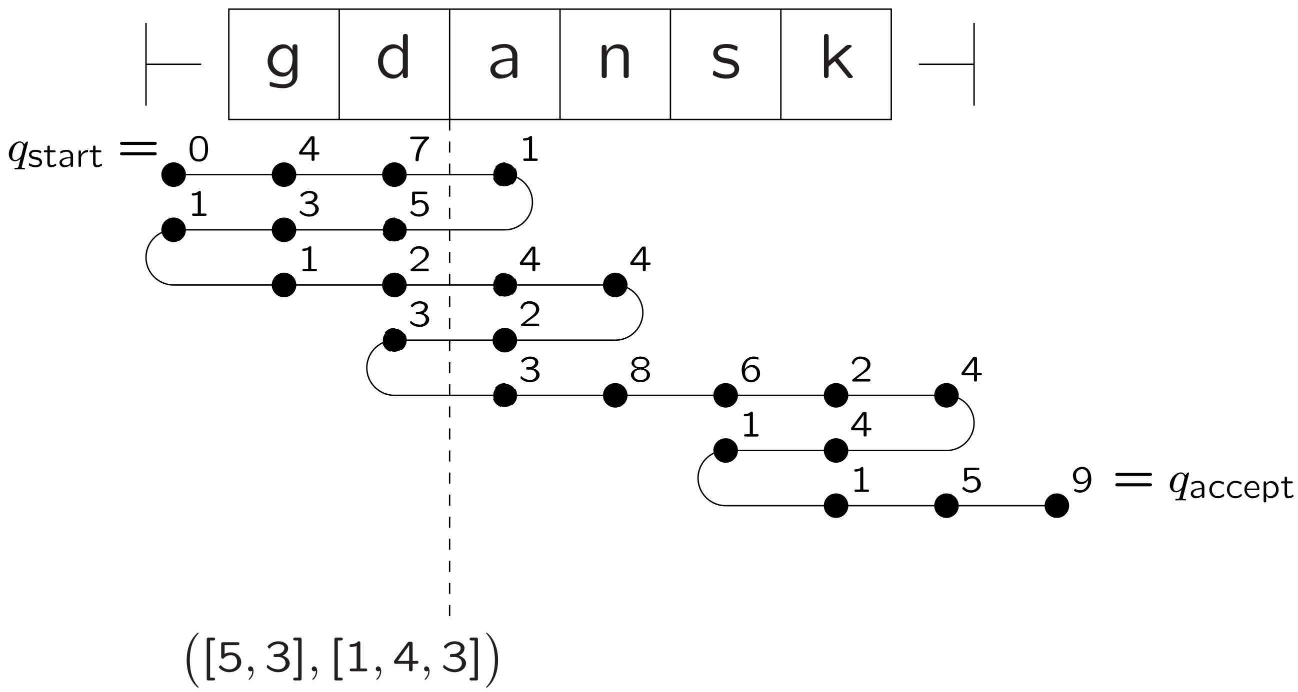
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



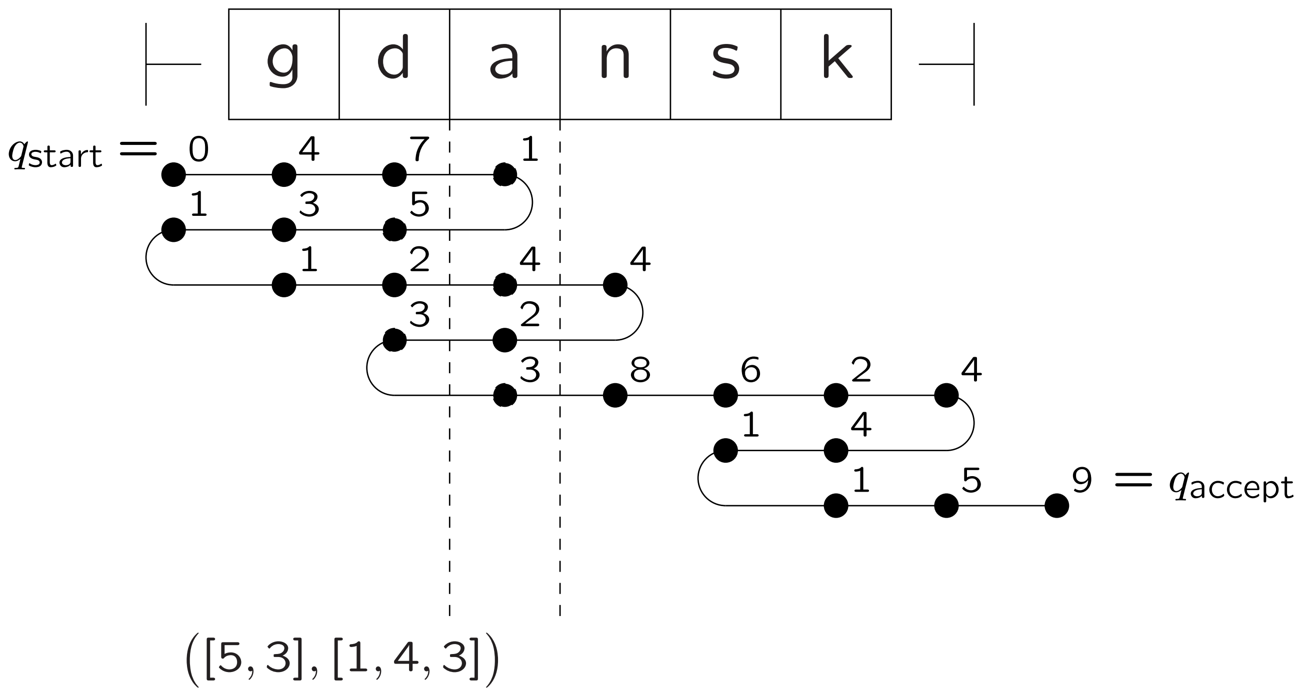
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



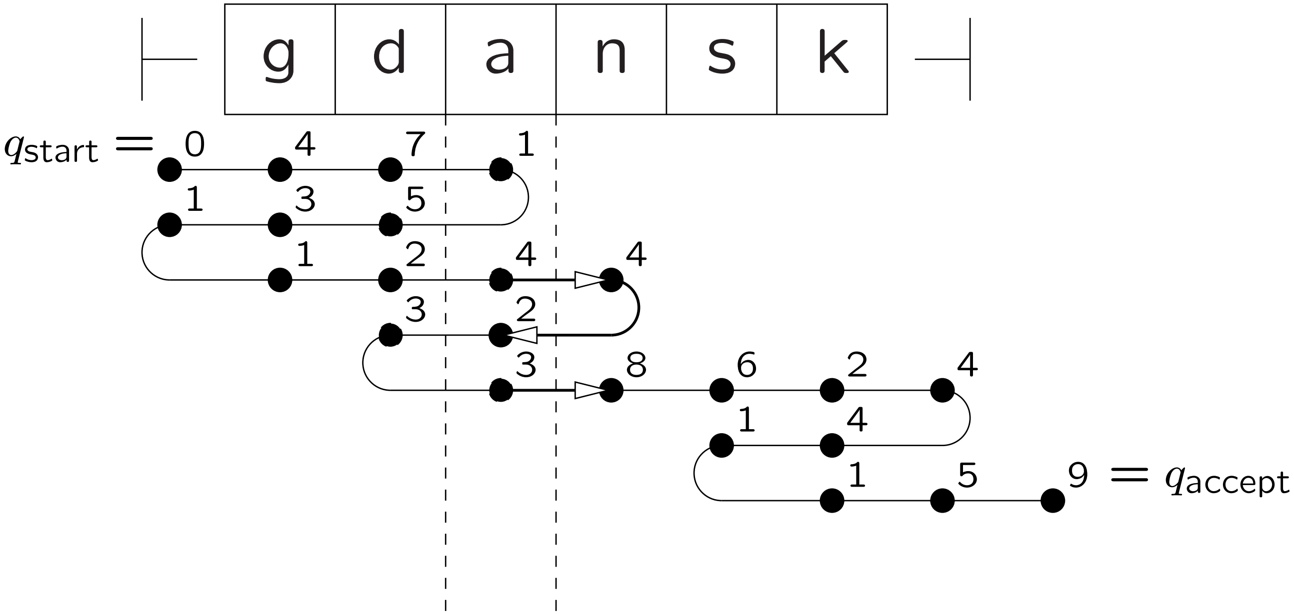
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

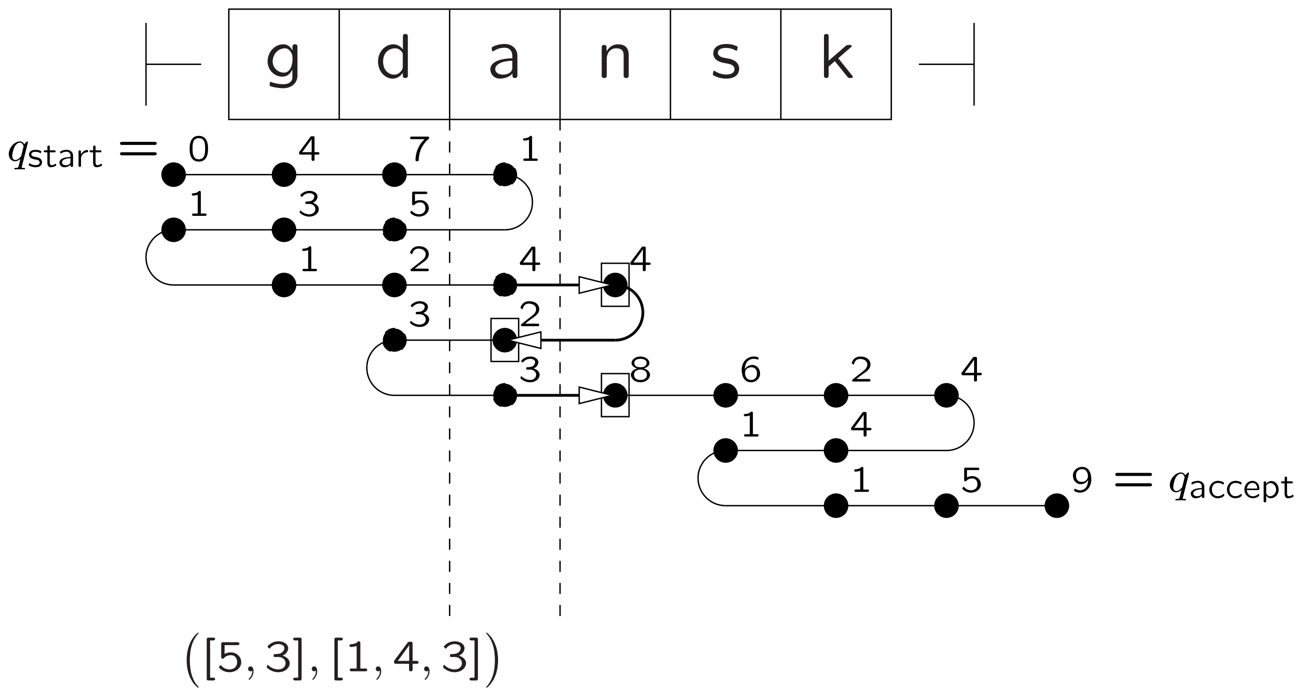


EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

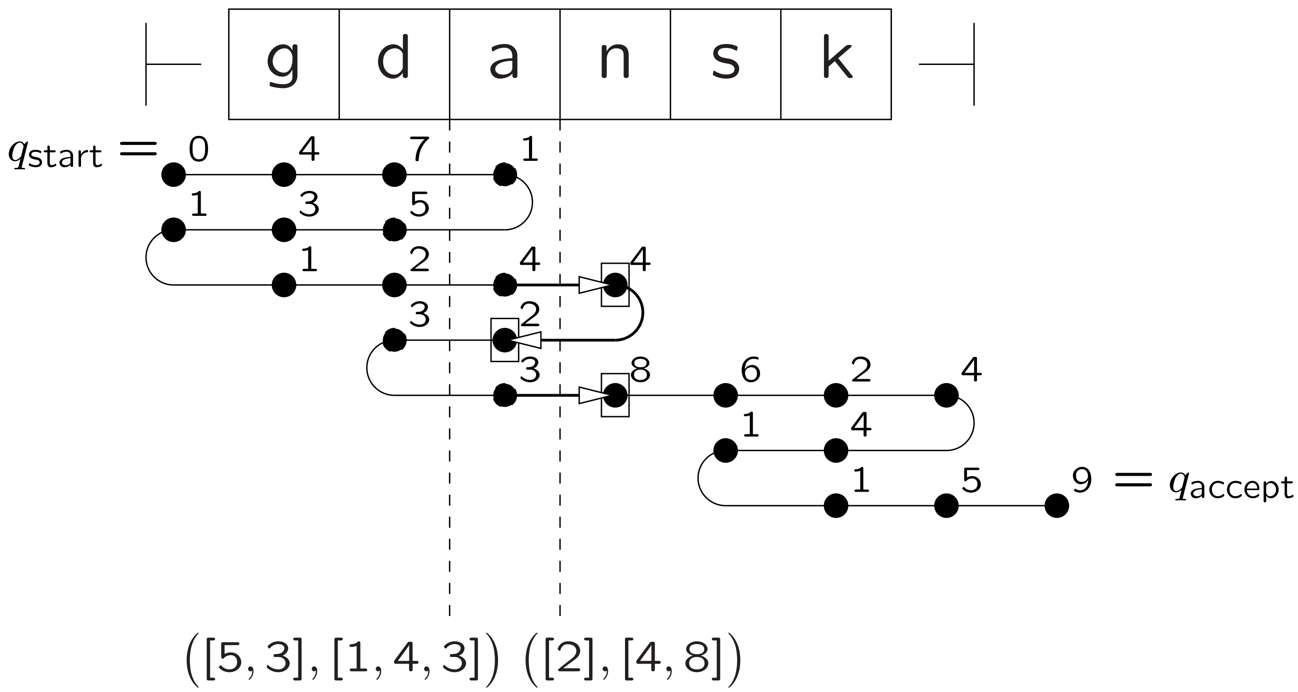


$\big([5,3],[1,4,3]\big)$

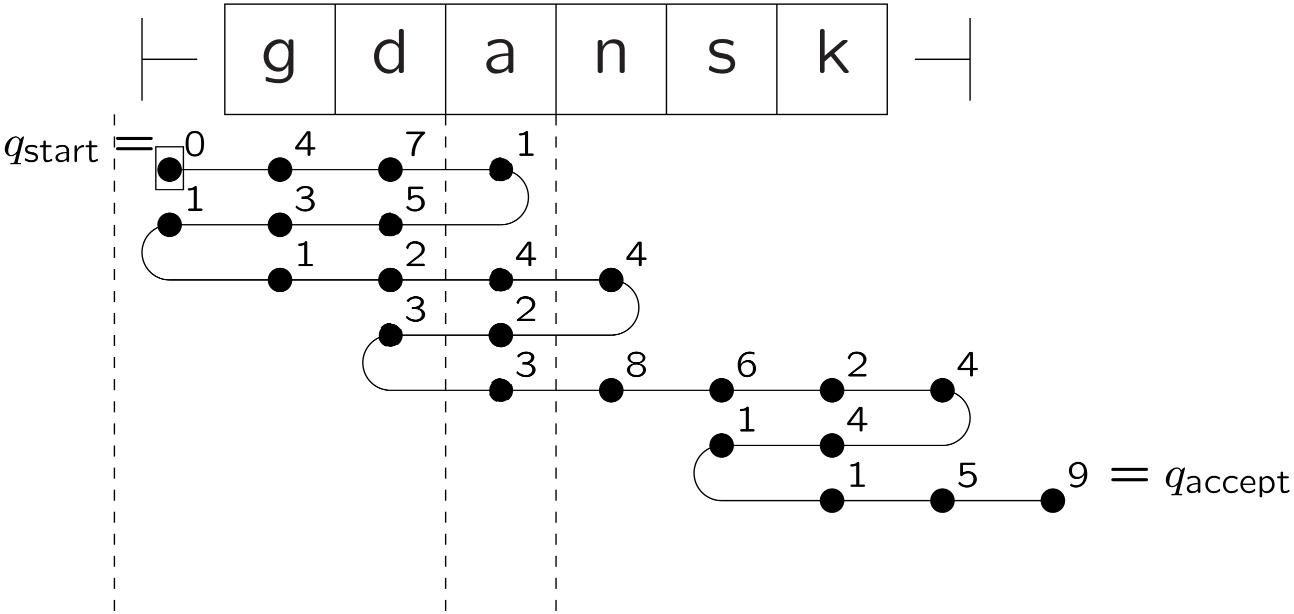
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

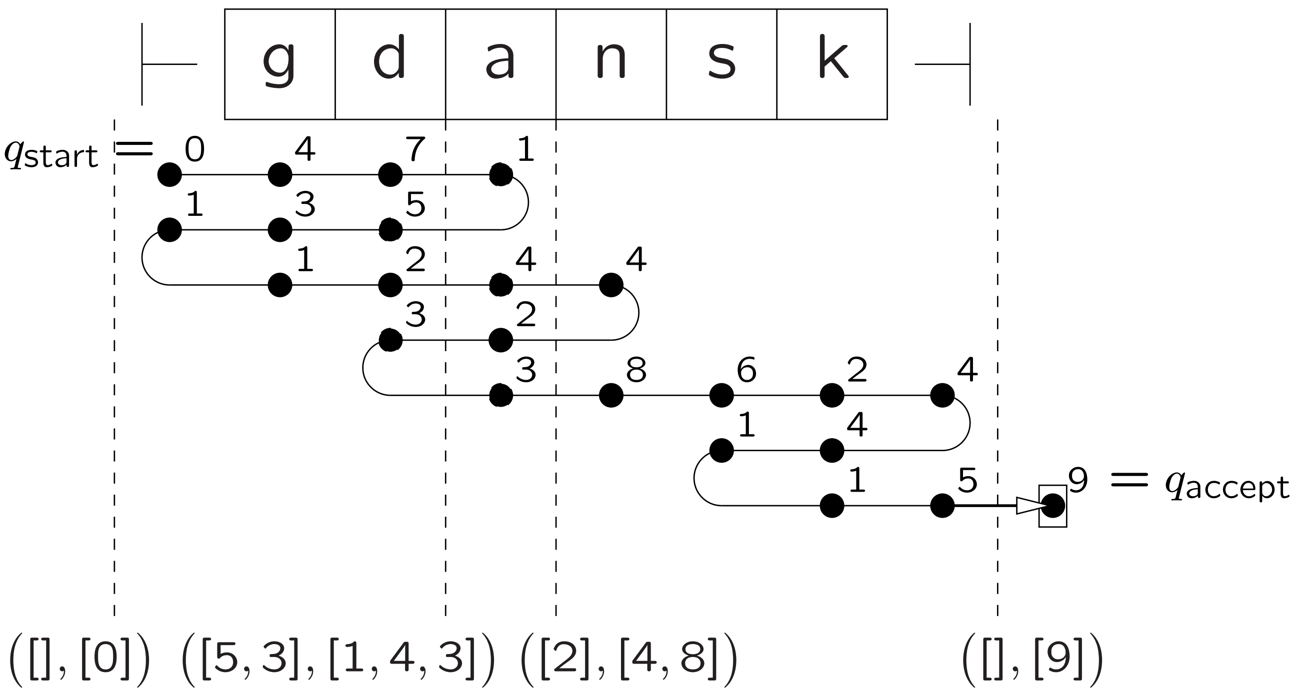


EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

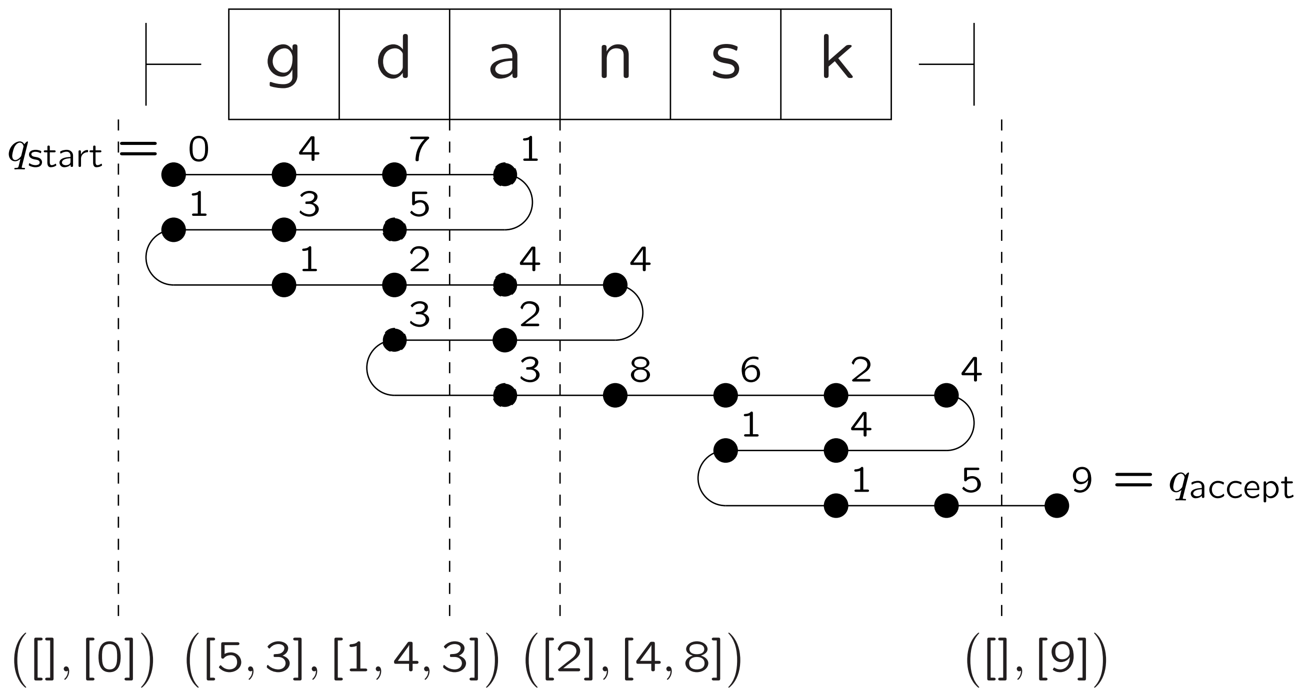


$([], [0])$ $([5, 3], [1, 4, 3])$ $([2], [4, 8])$

EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

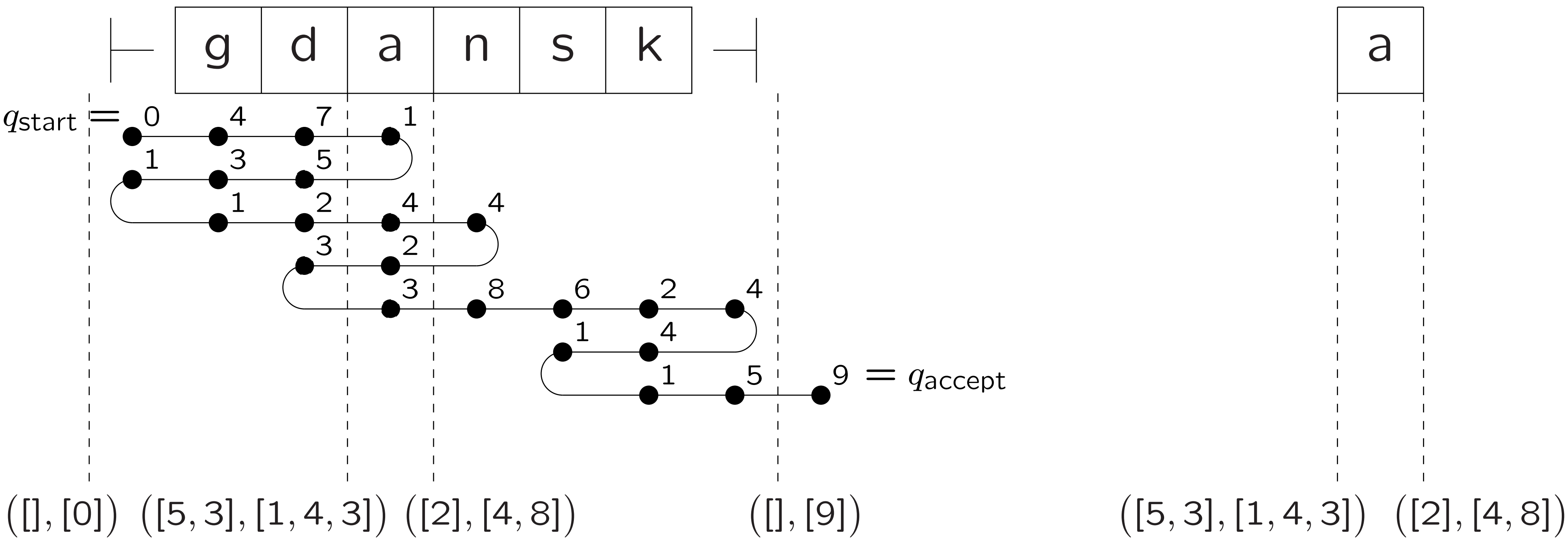


EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

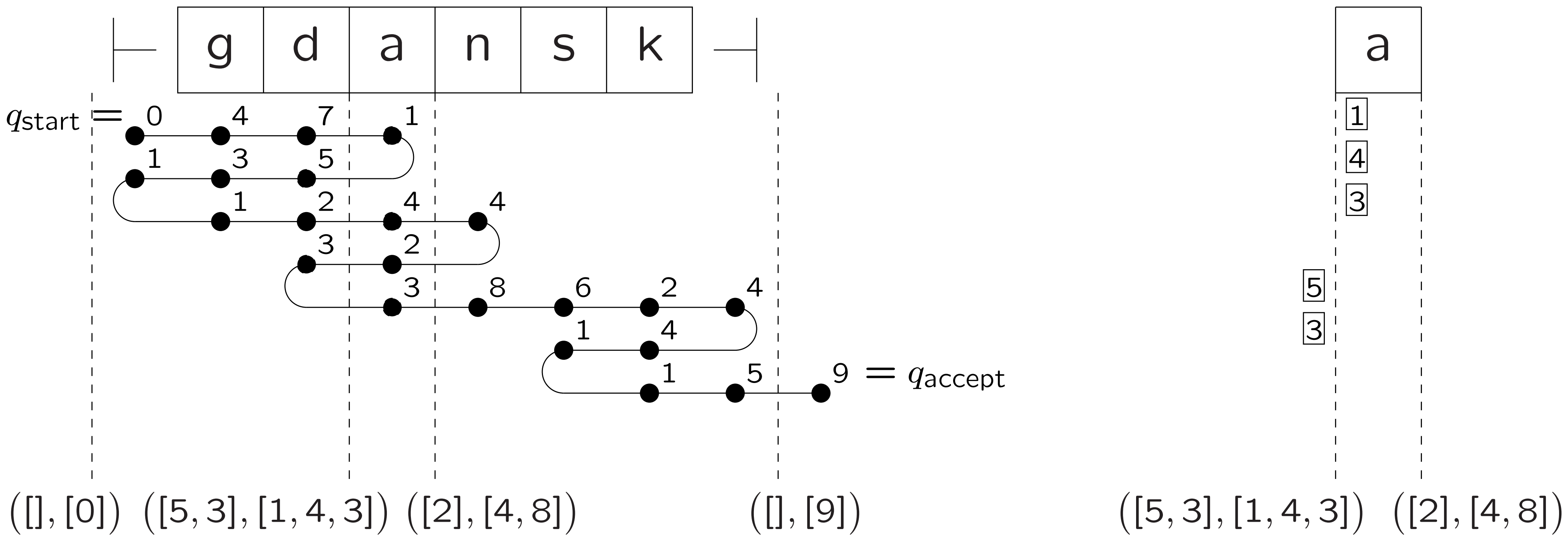


CROSSING-SEQUENCE CONSTRUCTION

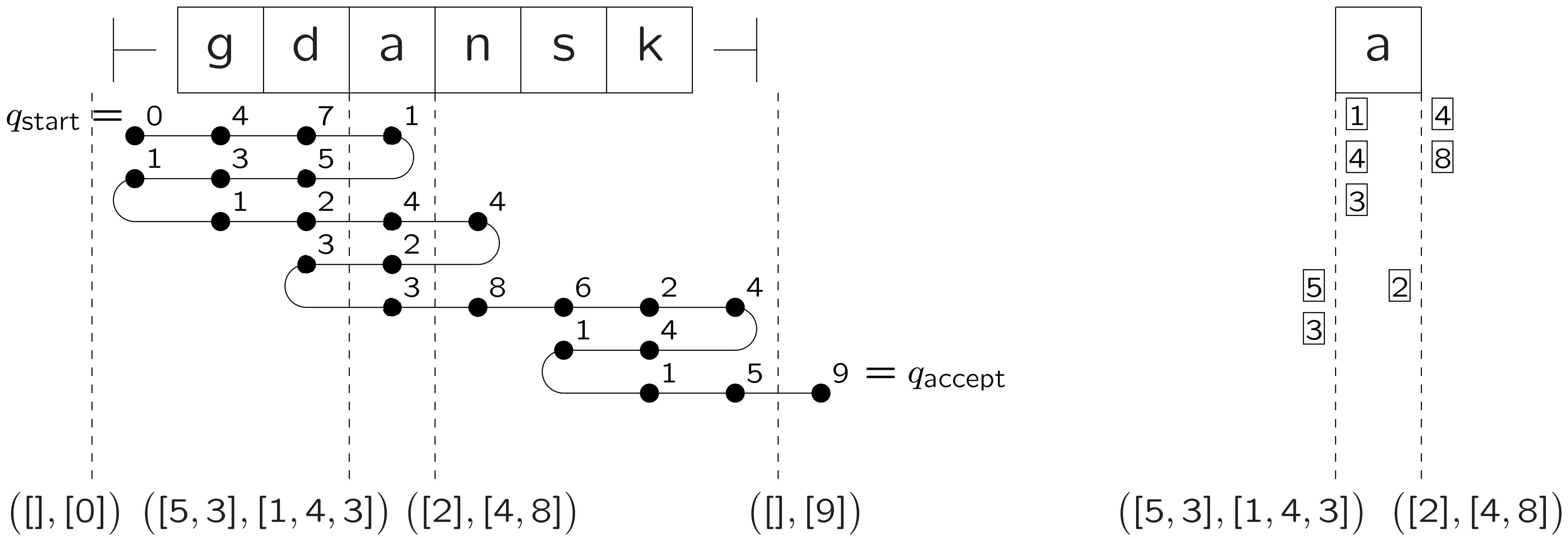
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



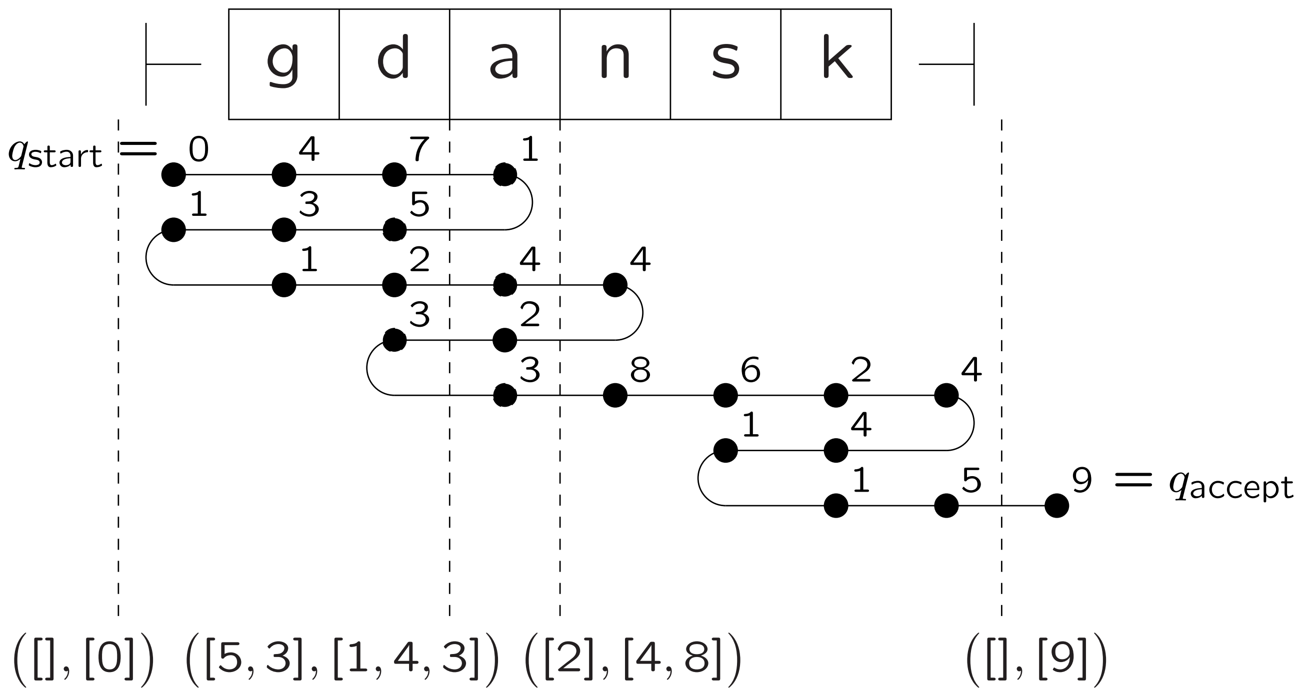
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

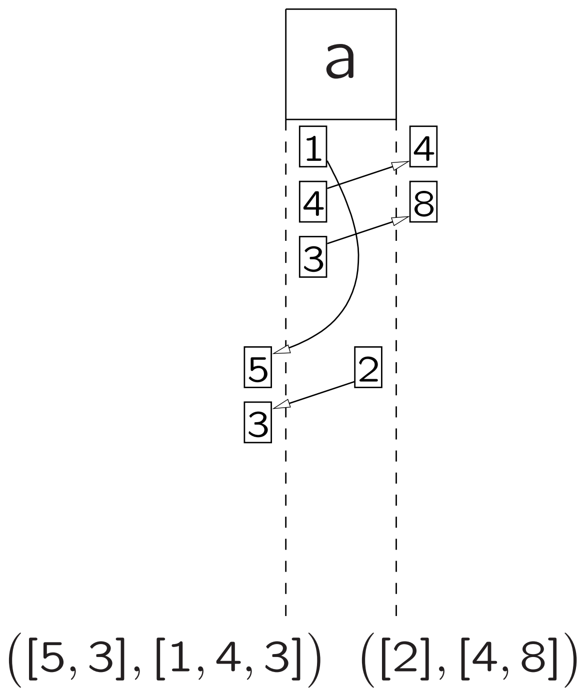


EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

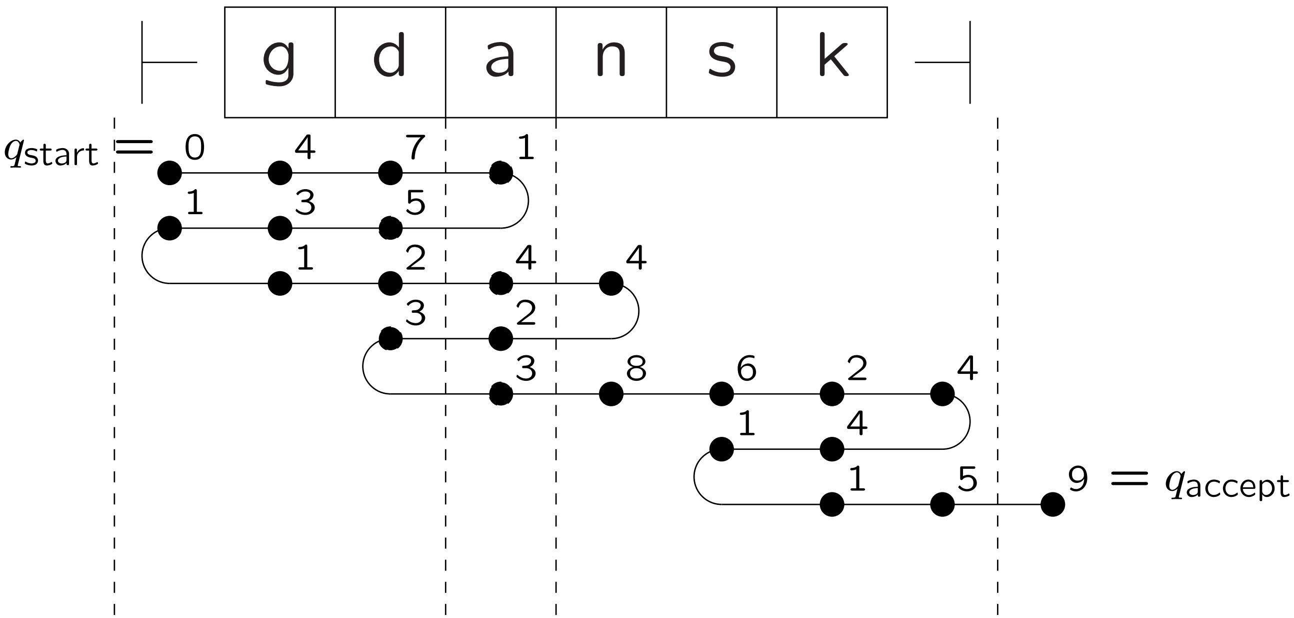


EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

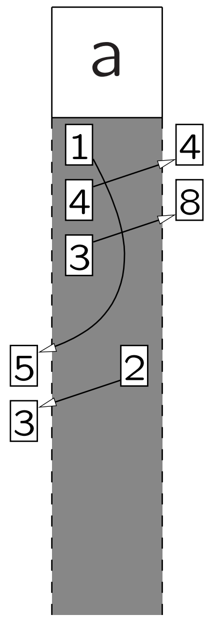




EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



$([], [0])$ $([5,3], [1,4,3])$ $([2], [4,8])$ $([], [9])$

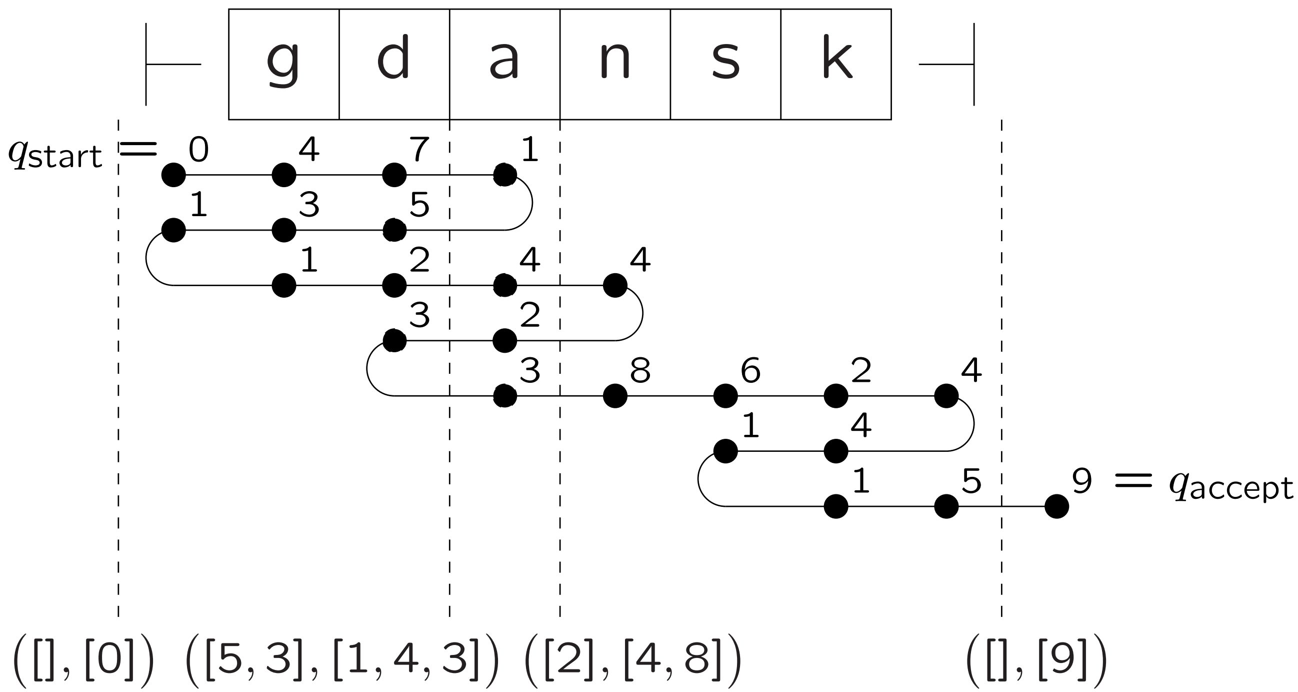


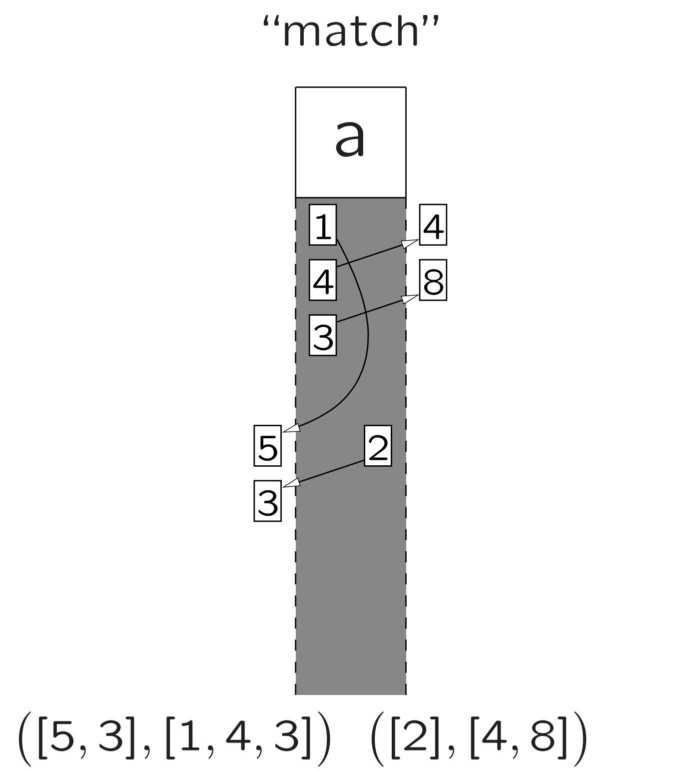
$([5,3], [1,4,3])$ $([2], [4,8])$

∃ bijection that respects
the trans.function & the order

CROSSING-SEQUENCE CONSTRUCTION

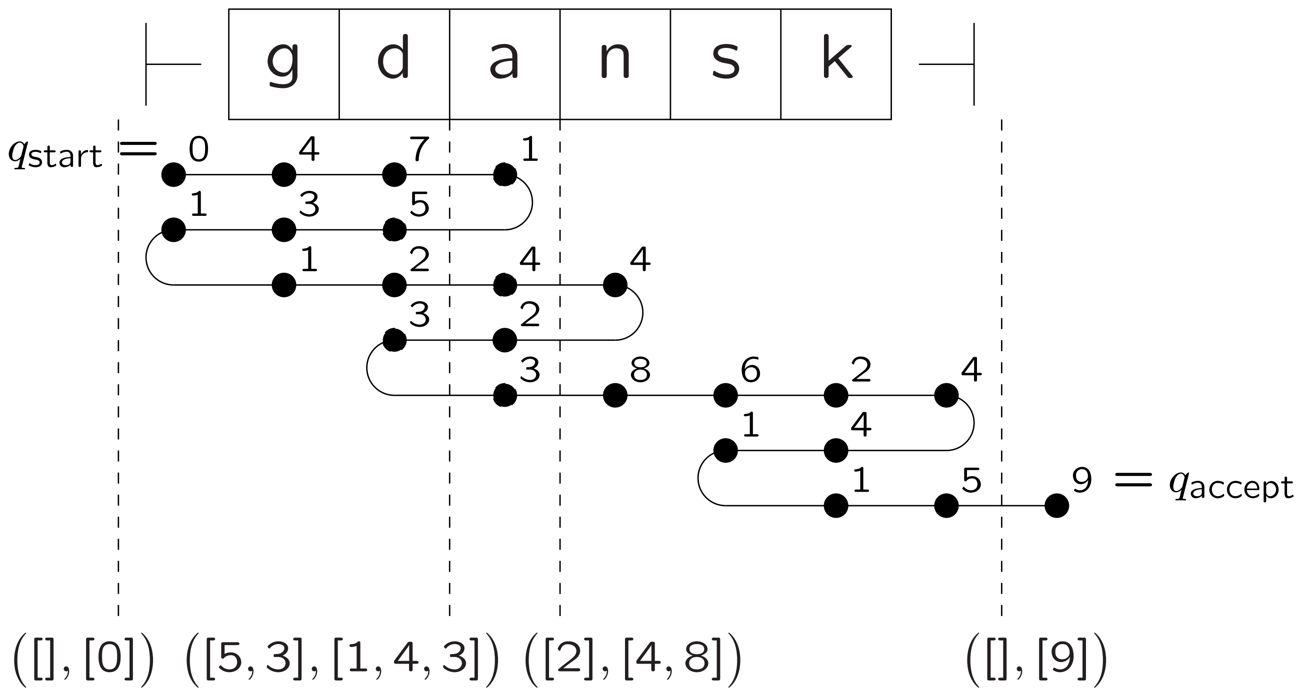
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

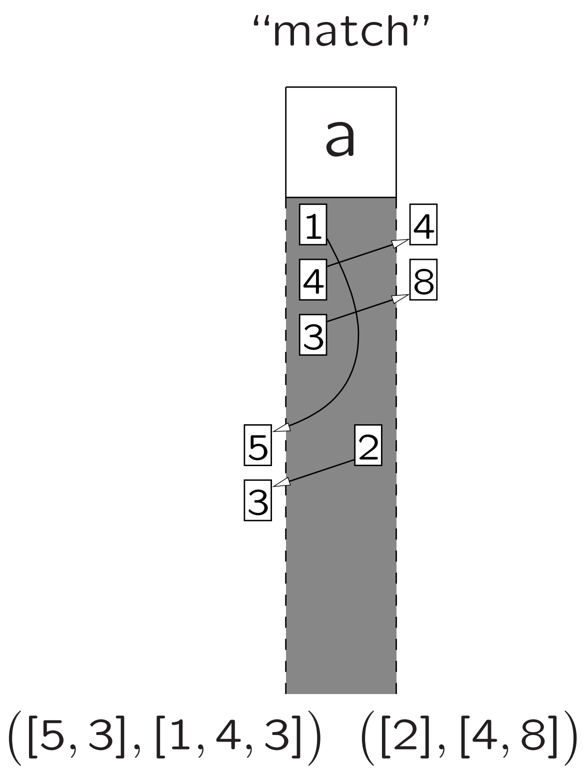




∃ bijection that respects
the trans.function & the order

EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.





∃ bijection that respects
the trans.function & the order

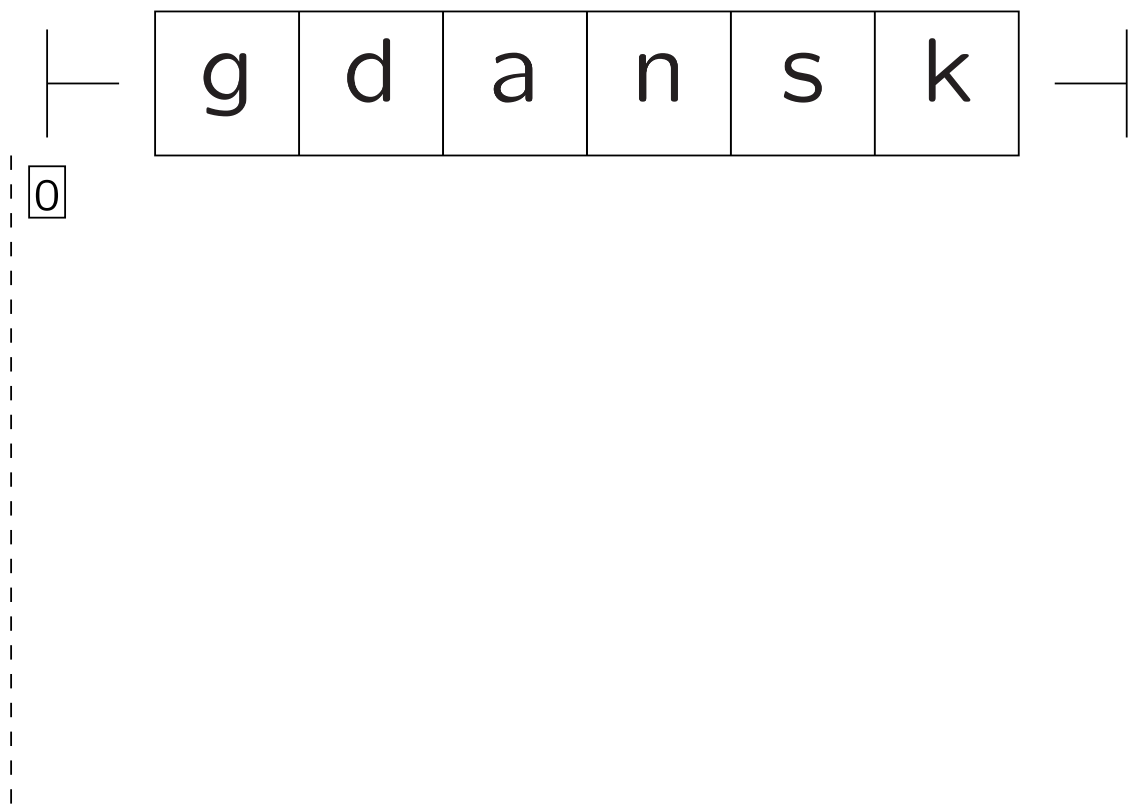
∃ list of crossing sequences from $([], [q_{\text{start}}])$ to $([], [q_{\text{accept}}])$ such that
every two successive of them match under the corresponding input symbol

EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\mathsf{start}} = 0$, $q_{\mathsf{accept}} = 9$.

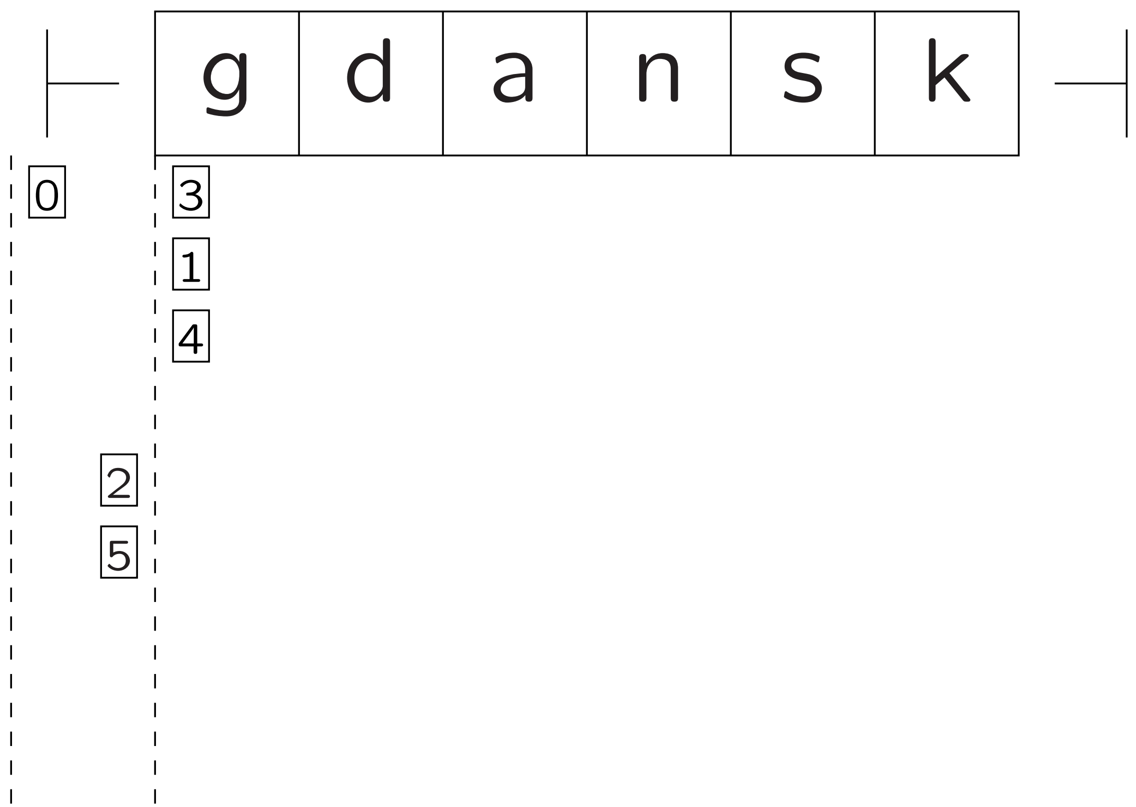
SIMULATING 1NFA:

states	=	all crossing-sequences of the 2NFA
start state	=	$\big([], [q_{\mathsf{start}}]\big)$
accept state	=	$\big([], [q_{\mathsf{accept}}]\big)$
$\delta\big(C, \mathsf{a}\big)$	=	{all crossing-sequences that match with C under a}

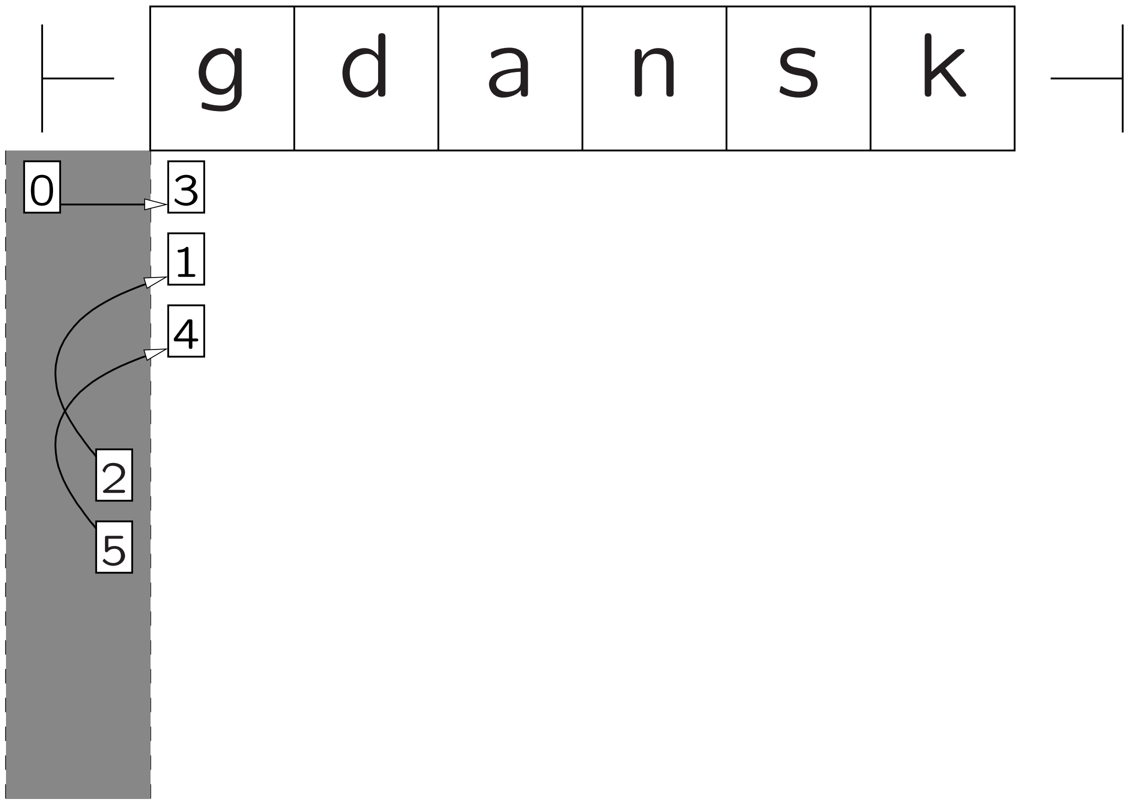
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



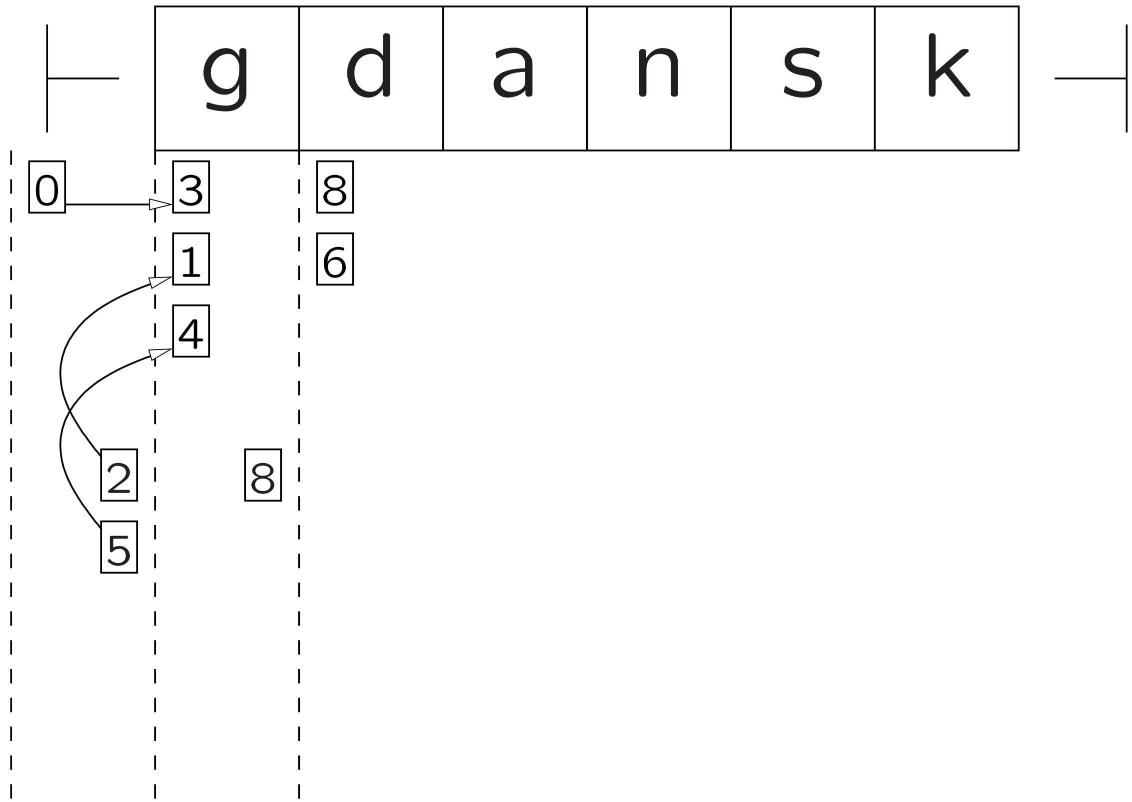
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



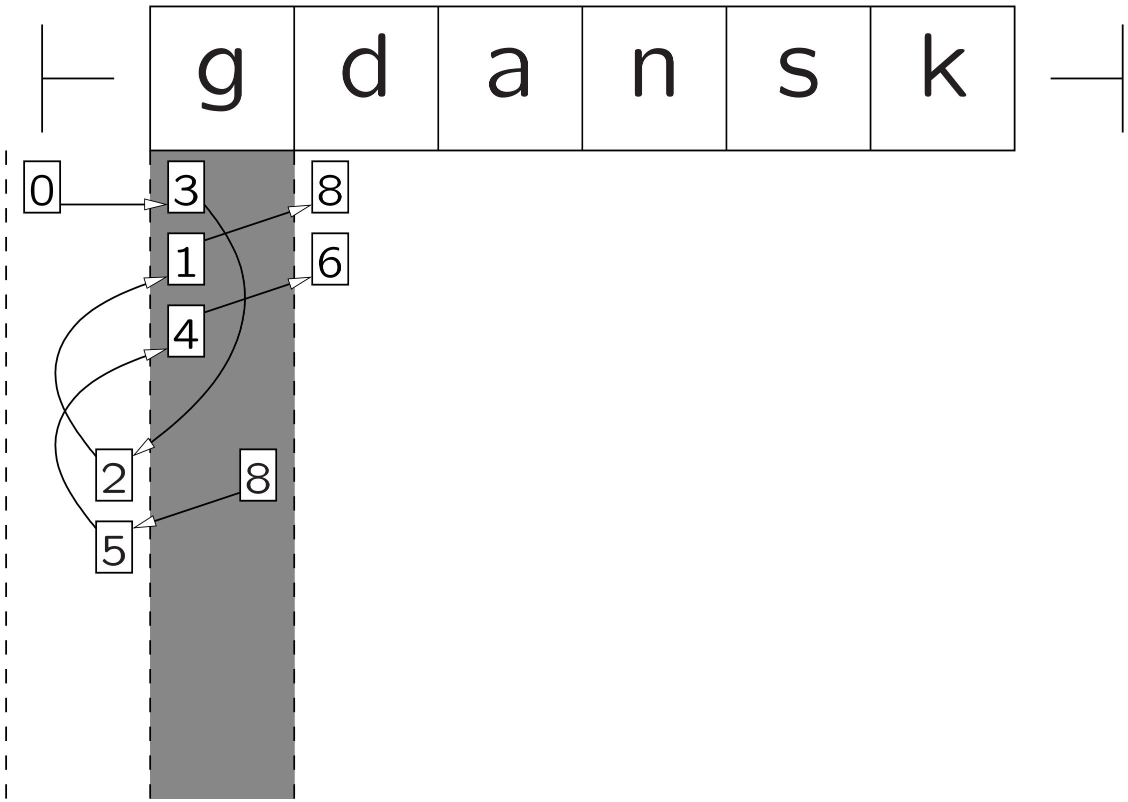
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, ..., 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

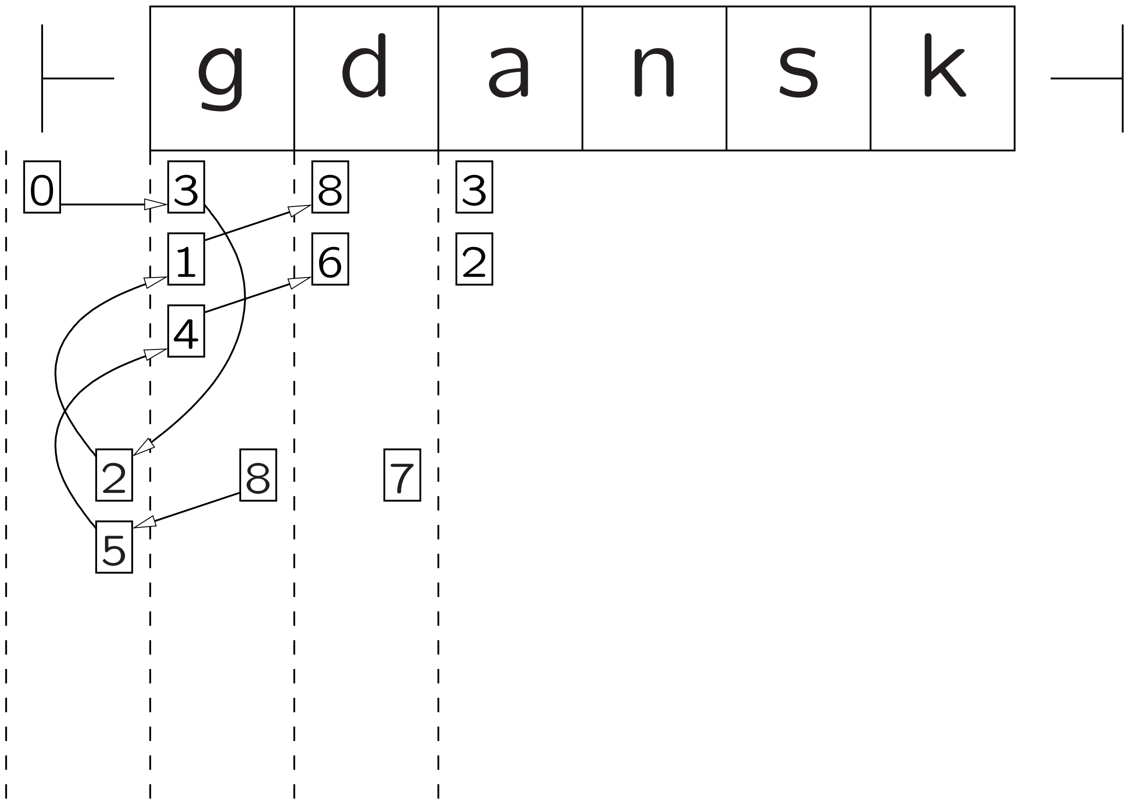


EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

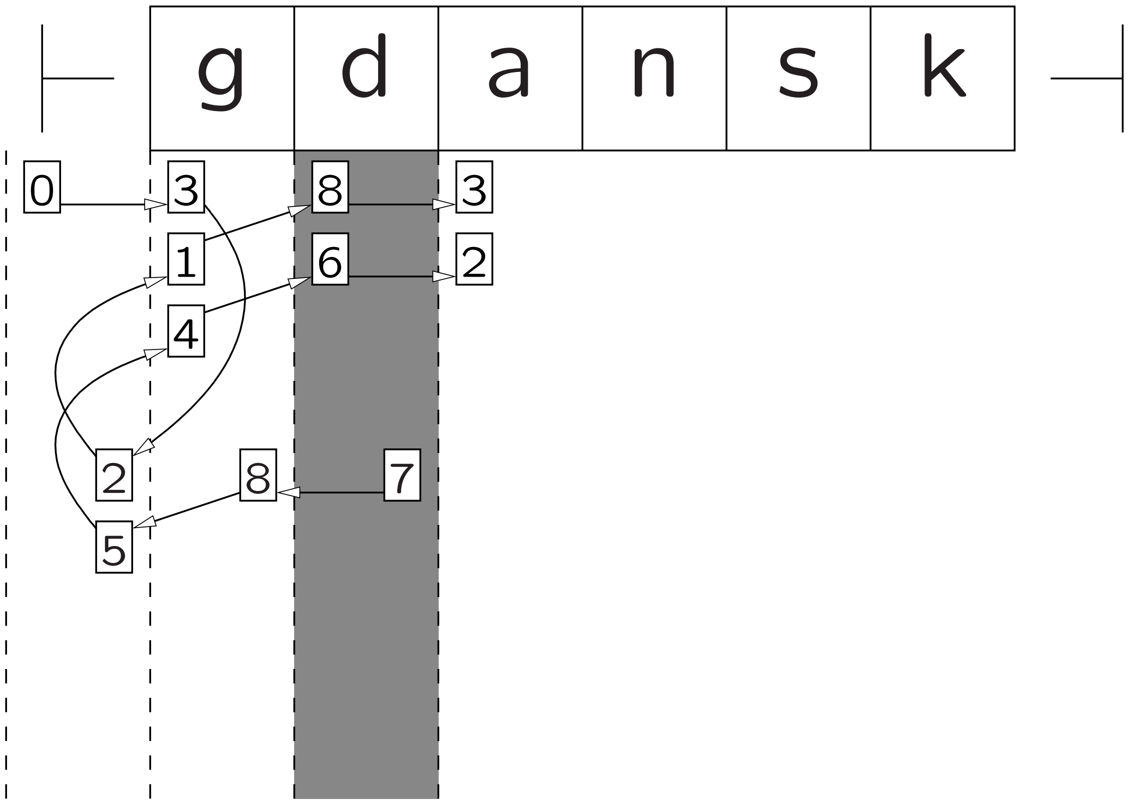


CROSSING-SEQUENCE CONSTRUCTION

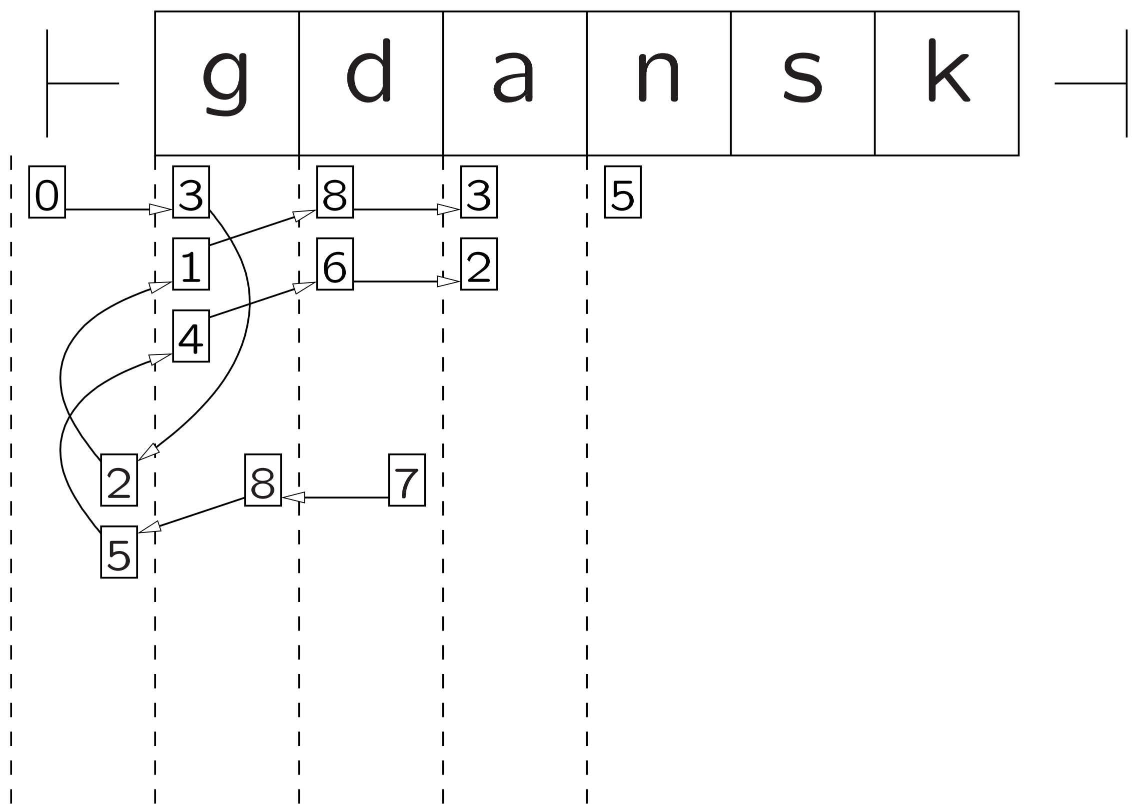
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



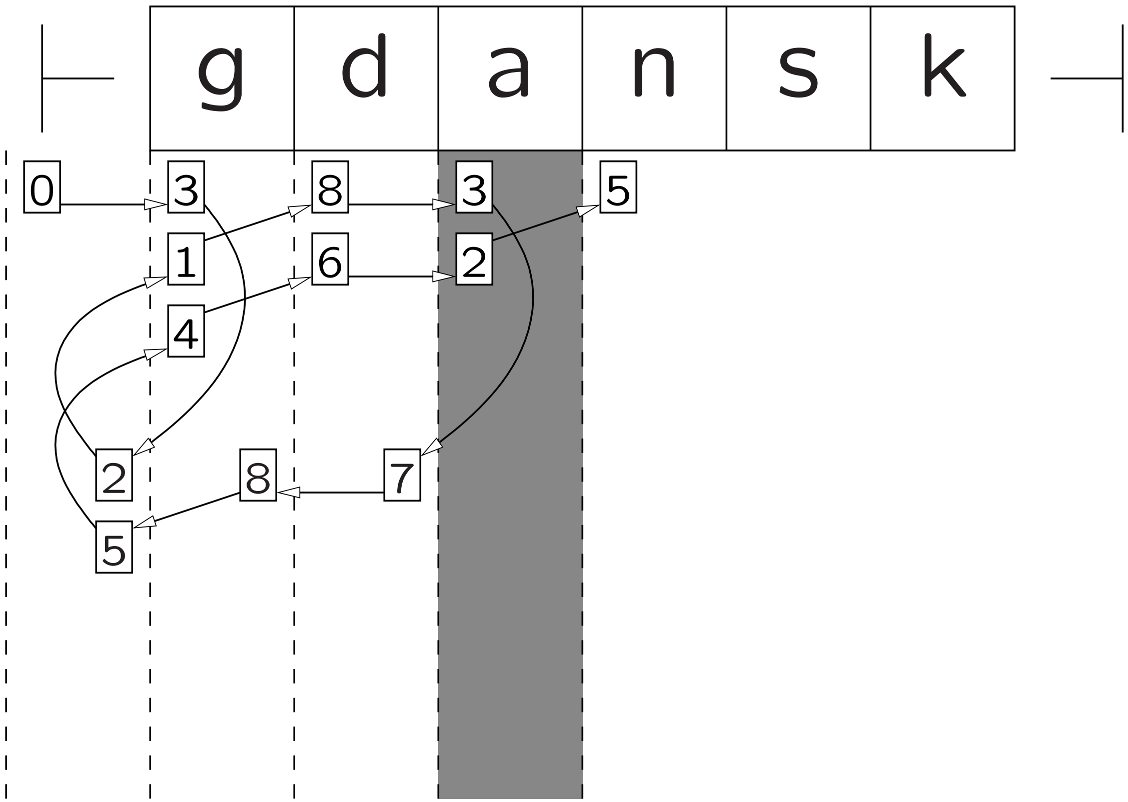
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



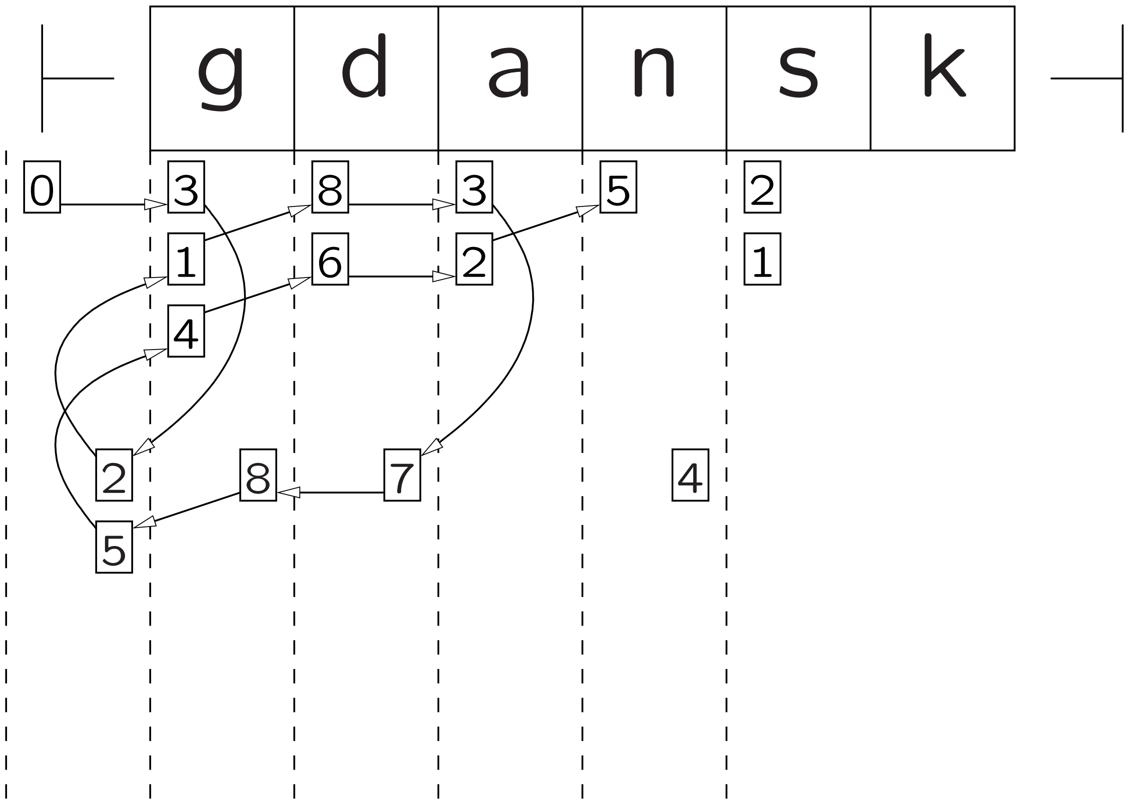
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



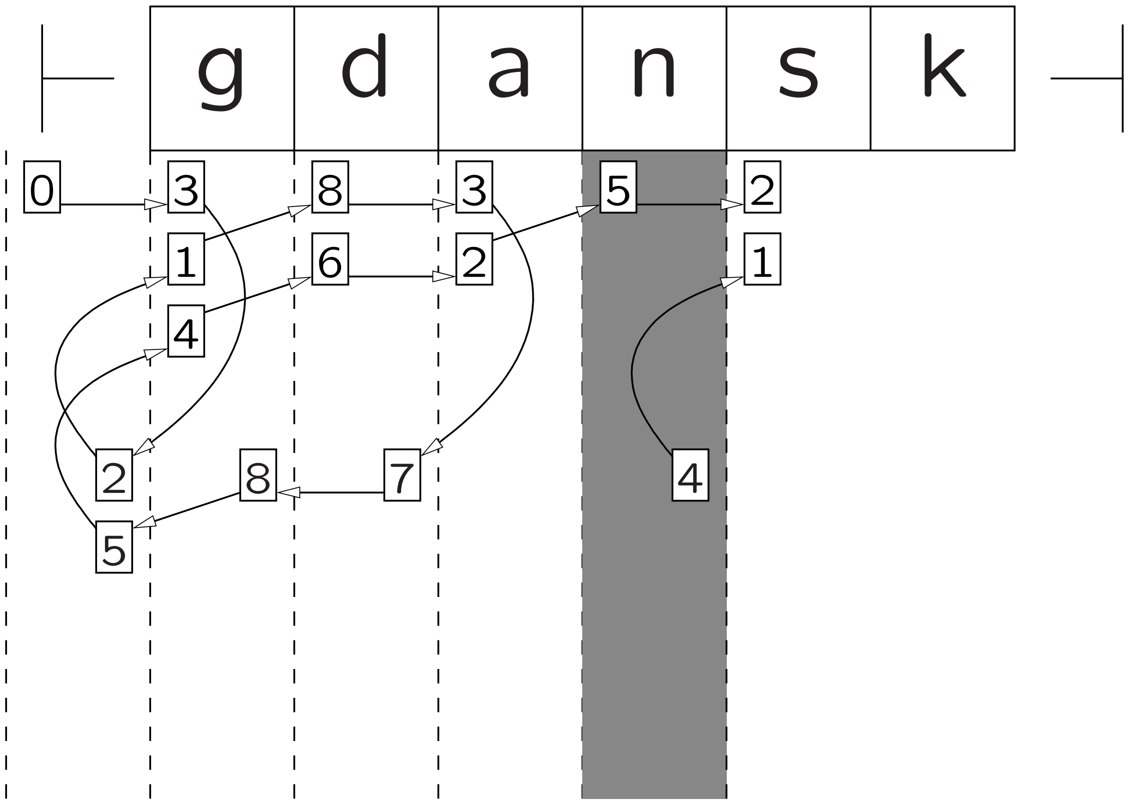
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



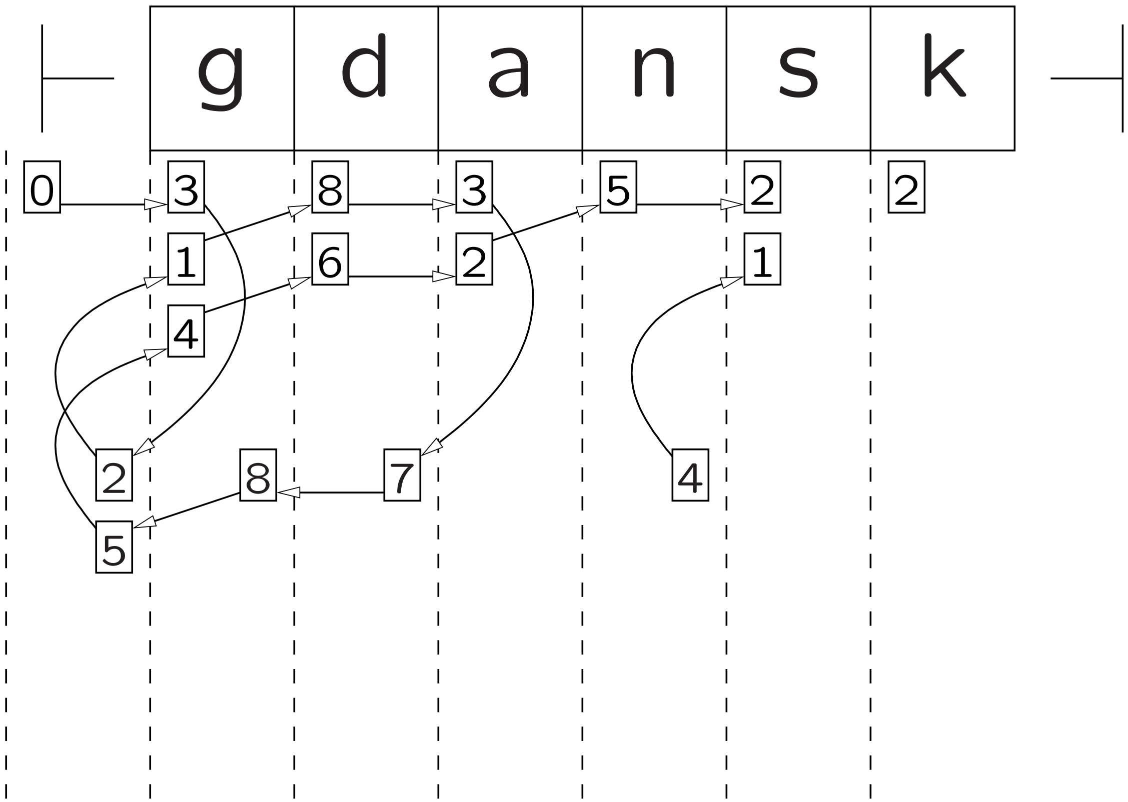
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



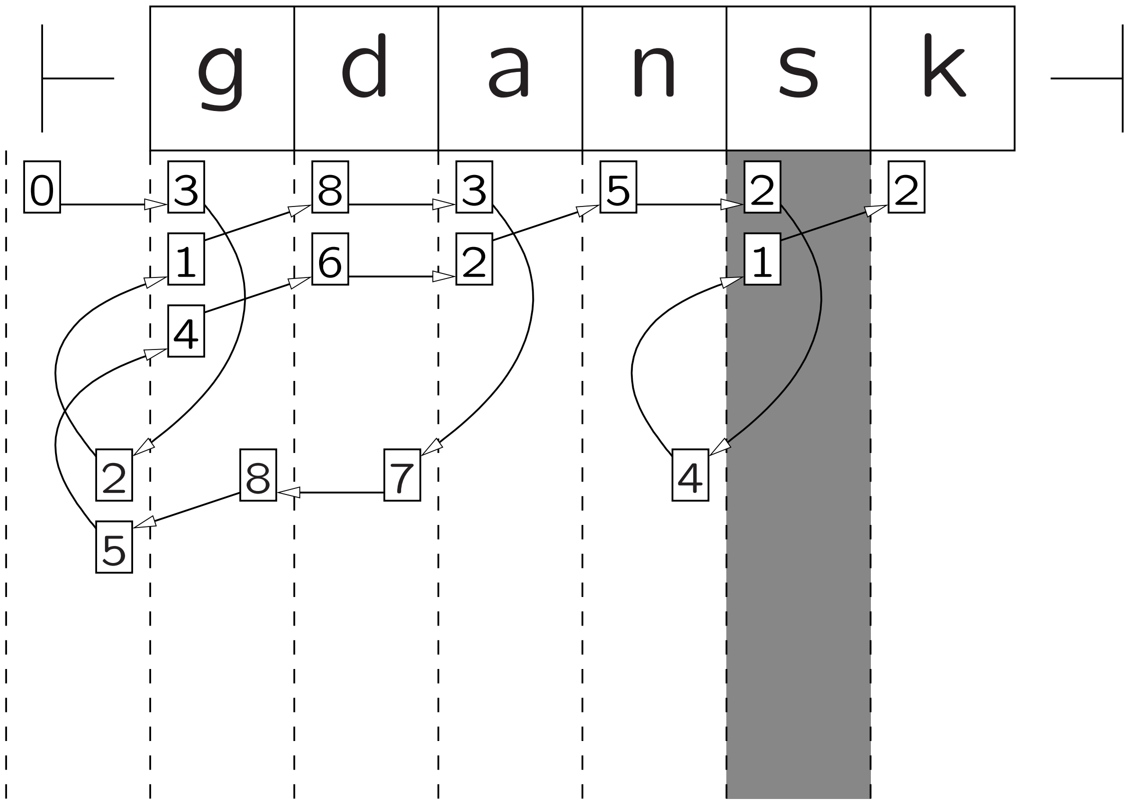
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

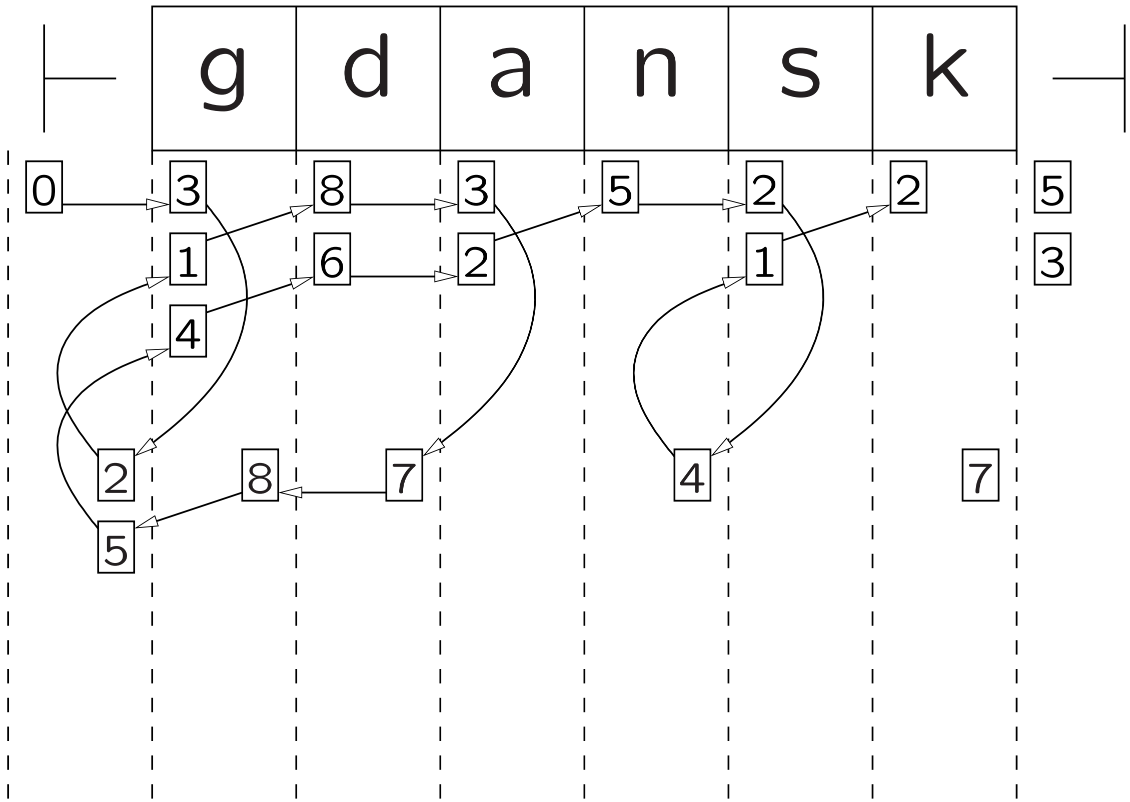


EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.

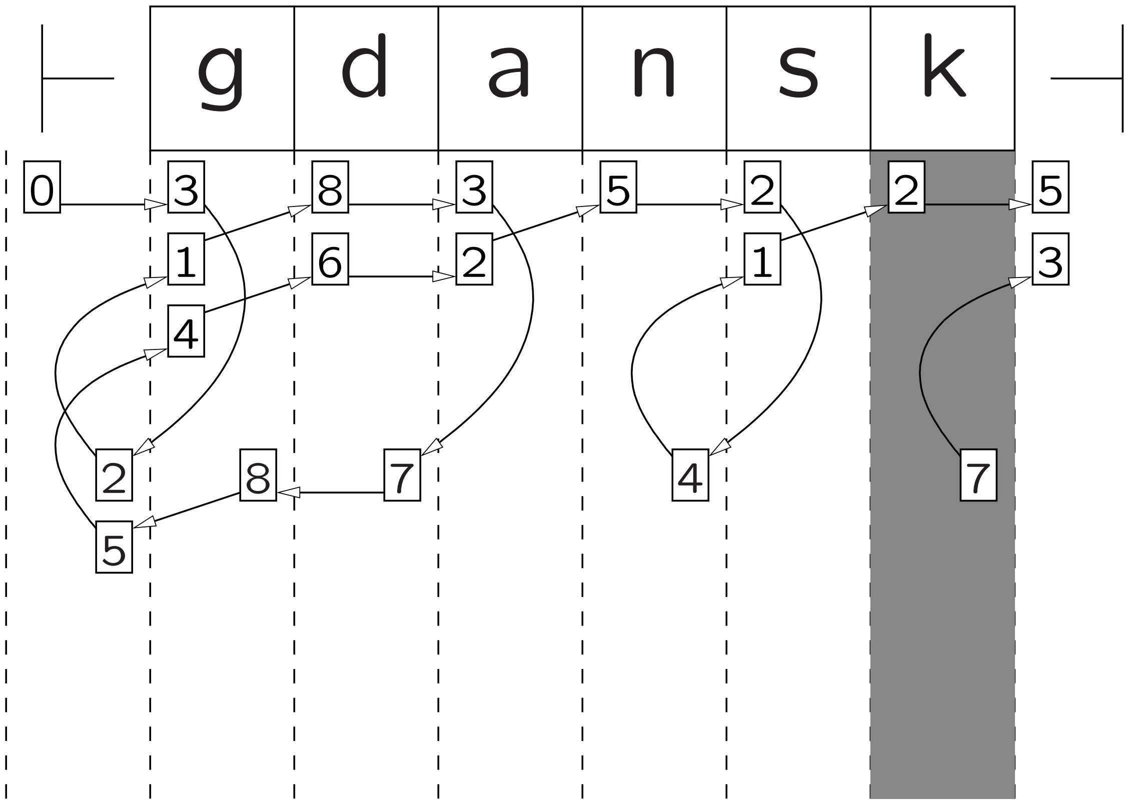


CROSSING-SEQUENCE CONSTRUCTION

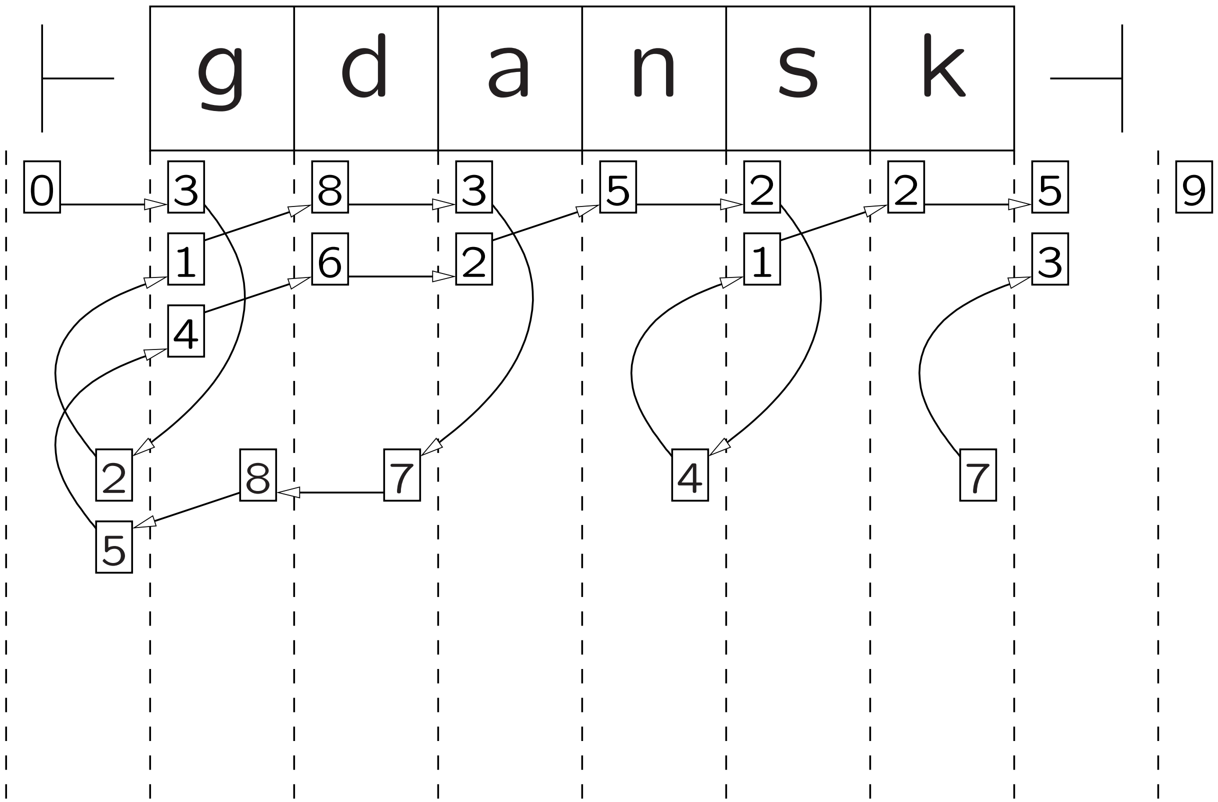
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



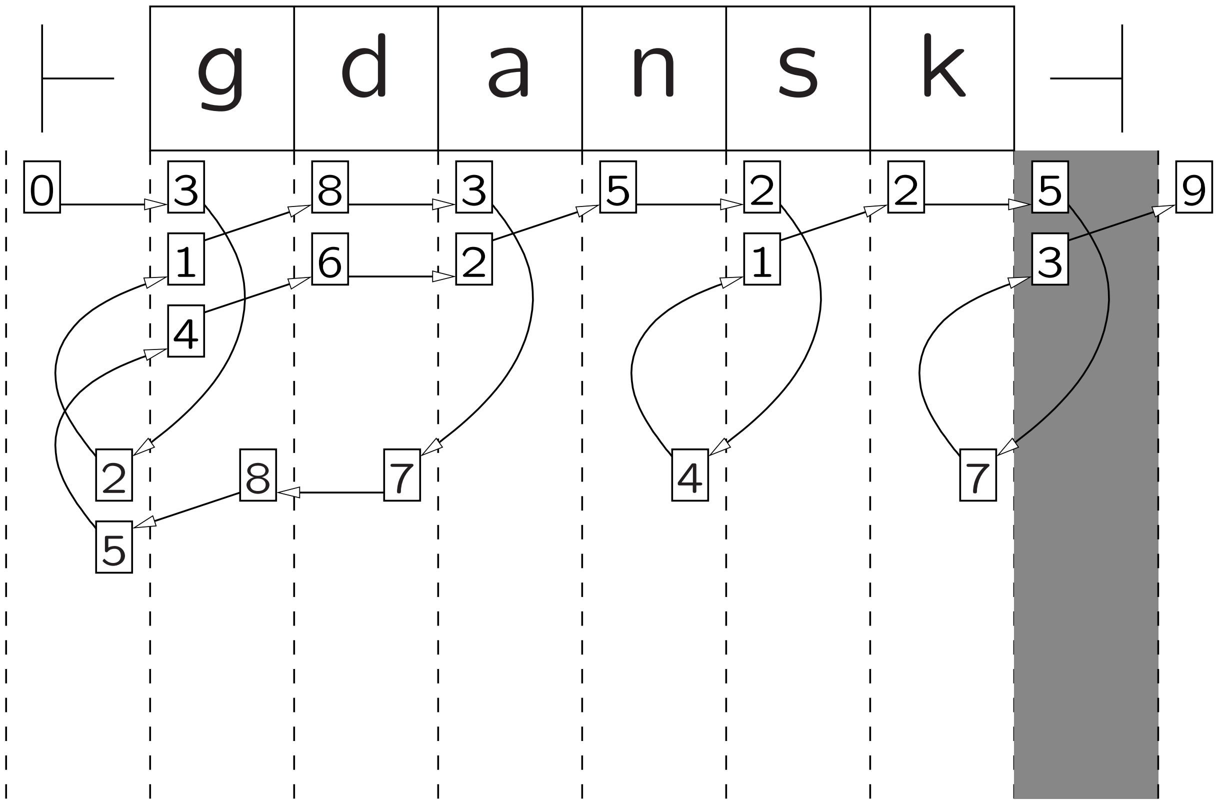
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



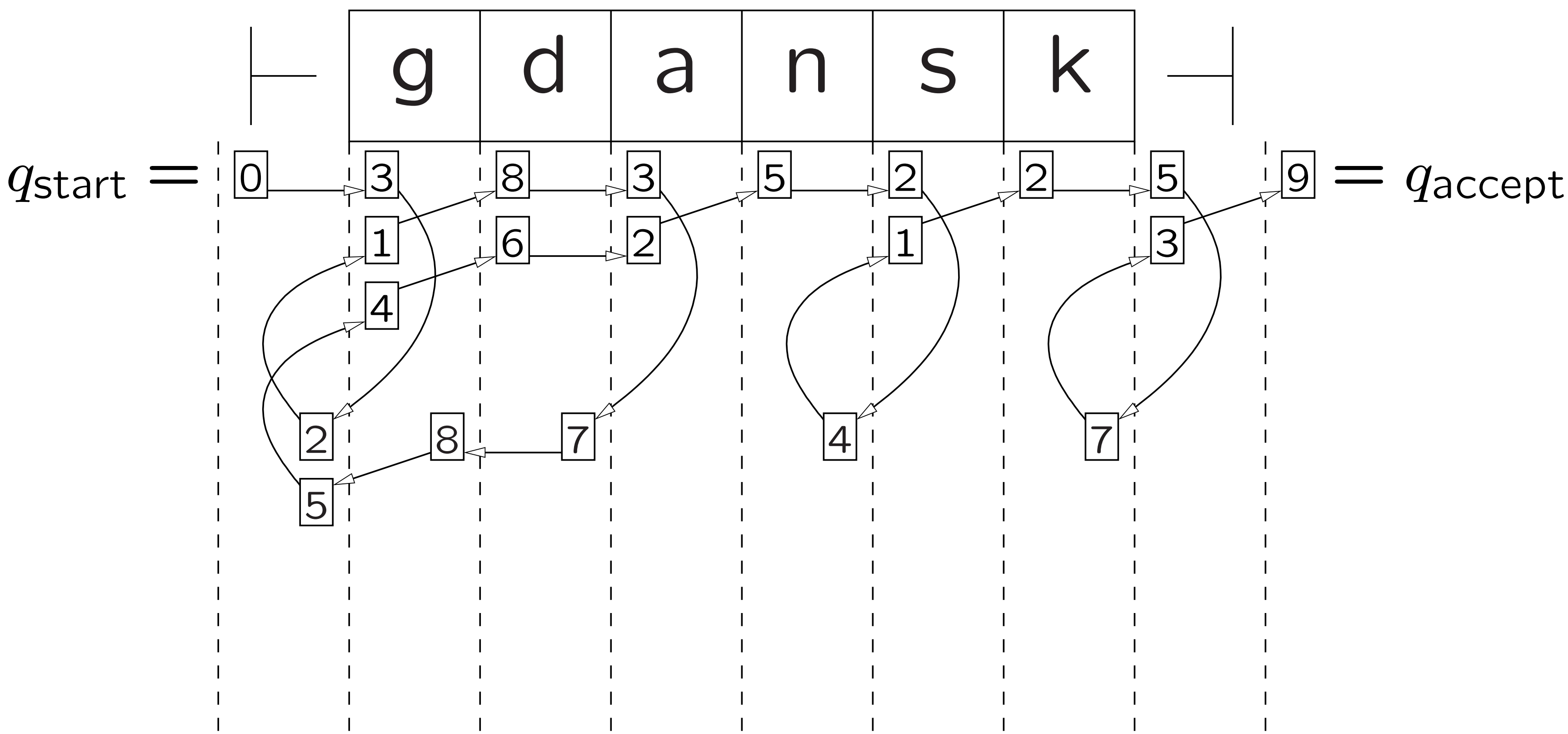
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



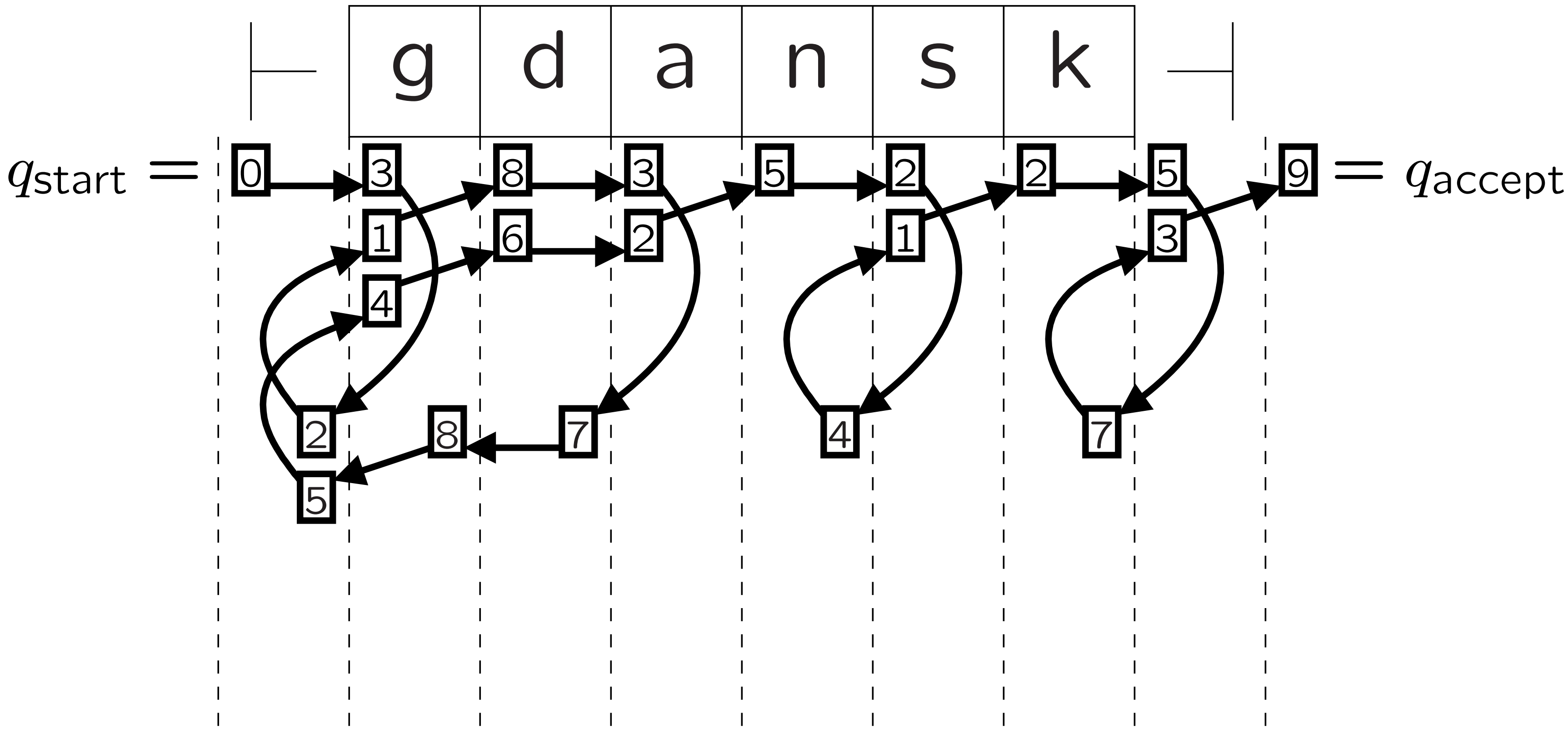
EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



EXAMPLE 2NFA: accepts names of beautiful cities, $Q = \{0, 1, 2, \ldots, 9\}$, $q_{\text{start}} = 0$, $q_{\text{accept}} = 9$.



CROSSING-SEQUENCE CONSTRUCTION

SIMULATING 1NFA:

states	$=$	all crossing-sequences of the 2NFA
start state	$=$	$\big([],[q_{\mathsf{start}}]\big)$
accept state	$=$	$\big([],[q_{\mathsf{accept}}]\big)$
$\delta\big(C,\mathsf{a}\big)$	$=$	{all crossing-sequences that match with C under a}

TOTAL SIZE: roughly $(n!)^2$

WHAT'S NEW?

WHAT'S NEW?

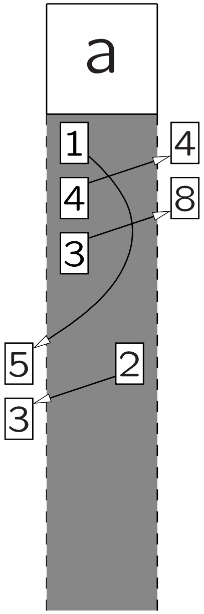
order is not important

order is not important

	CROSSING-SEQUENCE	**FRONTIER**
EXAMPLE:	$\big([5,3],[1,4,3]\big)$	$\big(\{3,5\},\{1,3,4\}\big)$
DEFINITION: (L,R) such that	$L,R \in Q^*$ & $\lvert L\rvert + 1 = \lvert R\rvert$	$L,R \subseteq Q$ & $\lvert L\rvert + 1 = \lvert R\rvert$
left half:	• which states? • in what order?	• which states?
right half:	• which states? (+1) • in what order?	• which states? (+1)

order is not important

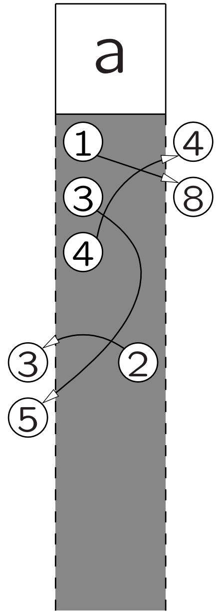
"match" of crossing-sequences



$\big([5,3],[1,4,3]\big)\ \ \big([2],[4,8]\big)$

$\exists$ bijection that respects
the trans.function & the order

"match" of **frontiers**



$\big(\{3,5\},\{1,3,4\}\big)\ \ \big(\{2\},\{4,8\}\big)$

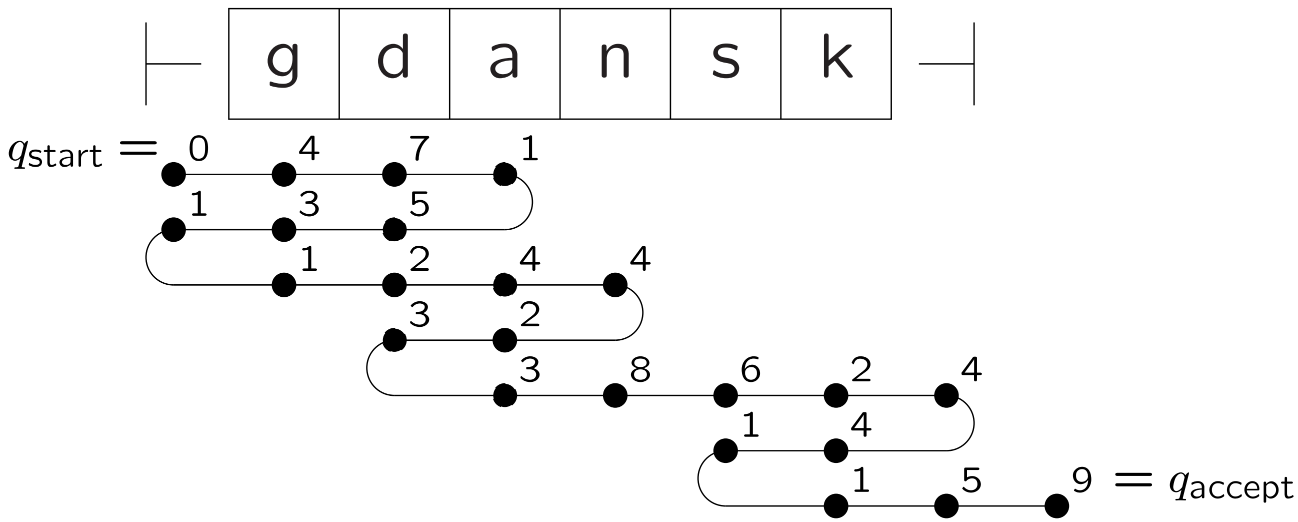
$\exists$ bijection that respects
the trans.function

FRONTIER CONSTRUCTION

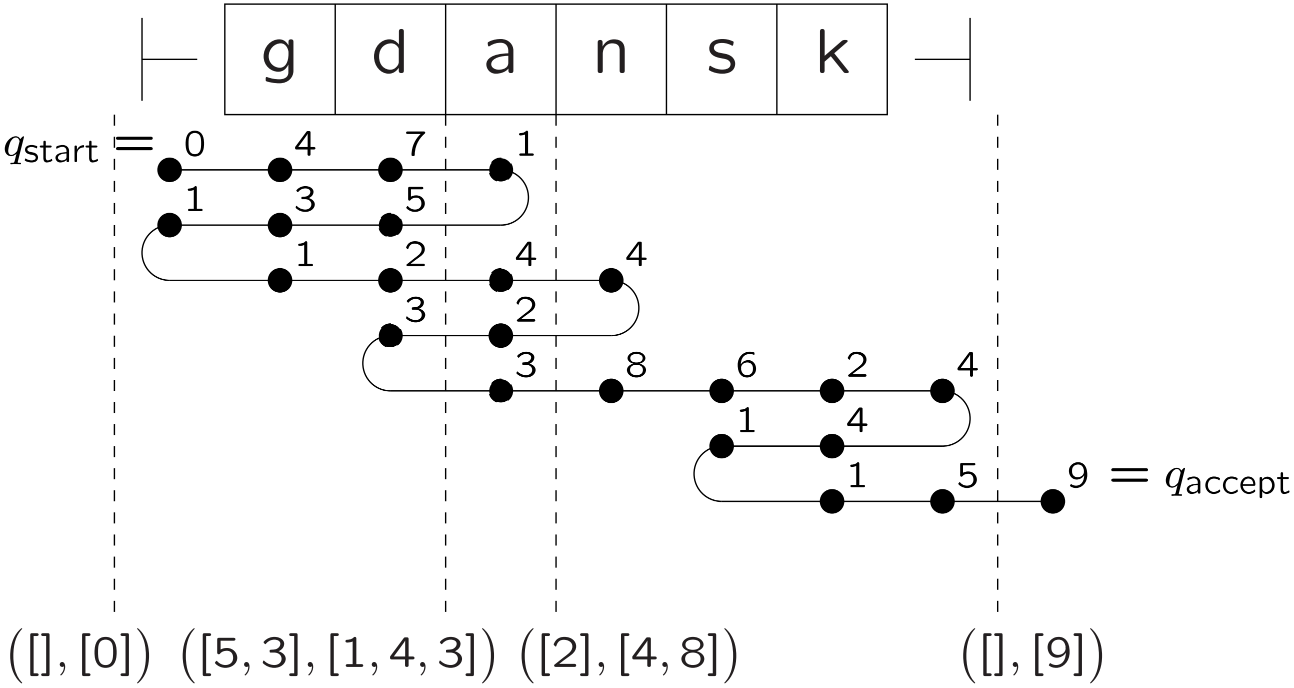
SIMULATING 1NFA:

states	=	all **frontiers** of the 2NFA
start state	=	$\big(\emptyset, \{q_{\mathsf{start}}\}\big)$
accept state	=	$\big(\emptyset, \{q_{\mathsf{accept}}\}\big)$
$\delta\big(F, \mathsf{a}\big)$	=	{all **frontiers** that match with F under a}

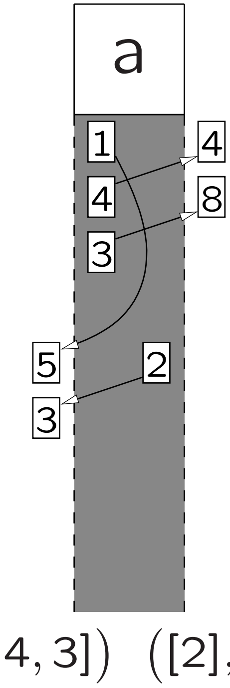
2NFA accepts ⇒ 1NFA accepts



2NFA accepts ⇒ 1NFA accepts



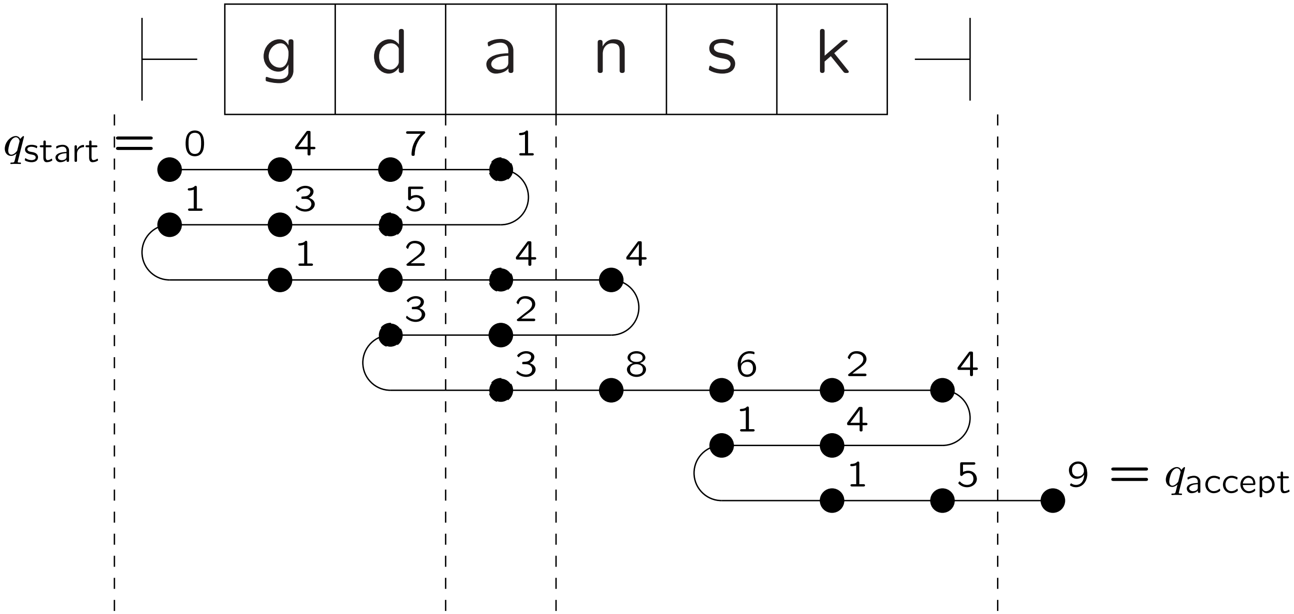
"match" of crossing-sequences



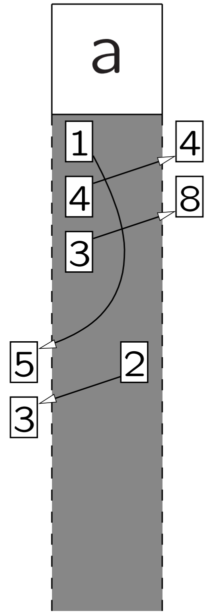
∃ bijection that respects
the trans.function & the order

2NFA accepts $\Rightarrow$ 1NFA accepts

“match” of crossing-sequences



$(\emptyset, \{0\})$ $(\{3,5\}, \{1,3,4\})$ $(\{2\}, \{4,8\})$ $(\emptyset, \{9\})$

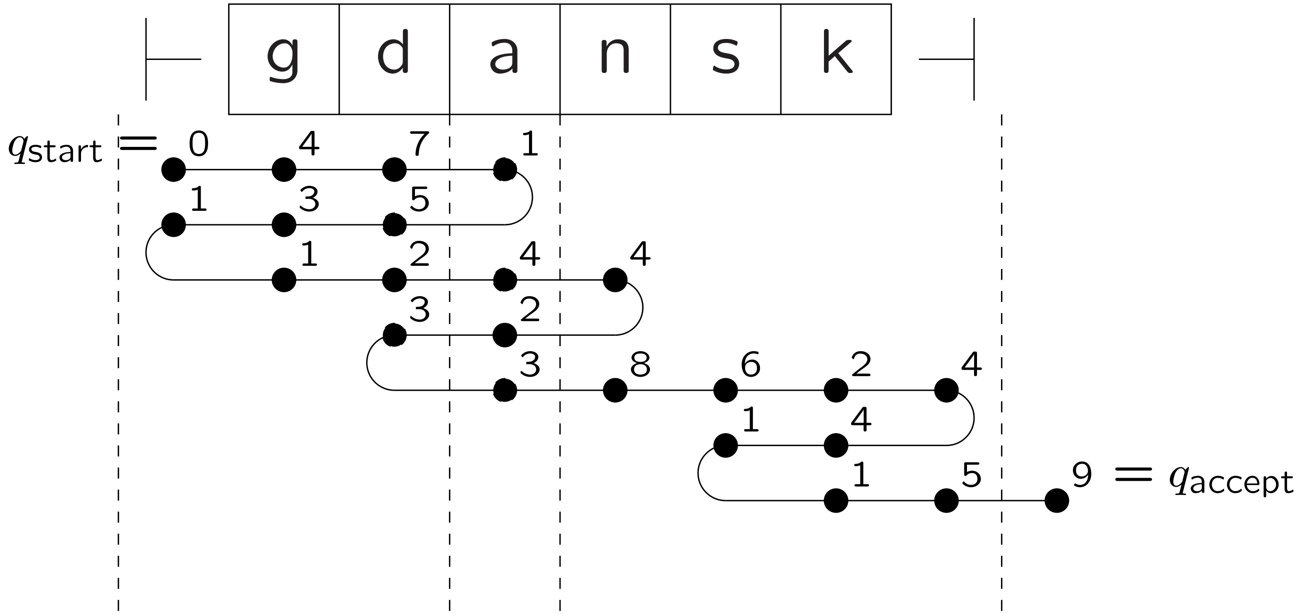


$([5,3],[1,4,3])$ $([2],[4,8])$

$\exists$ bijection that respects
the trans.function & the order

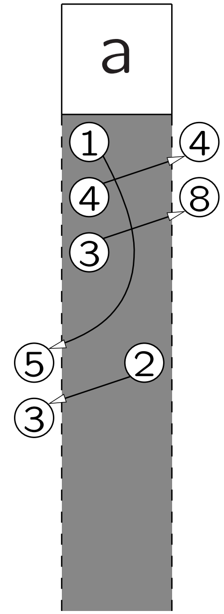
FRONTIER CONSTRUCTION

2NFA accepts ⇒ 1NFA accepts



$(\emptyset, \{0\})$ $(\{3,5\}, \{1,3,4\})$ $(\{2\}, \{4,8\})$ $(\emptyset, \{9\})$

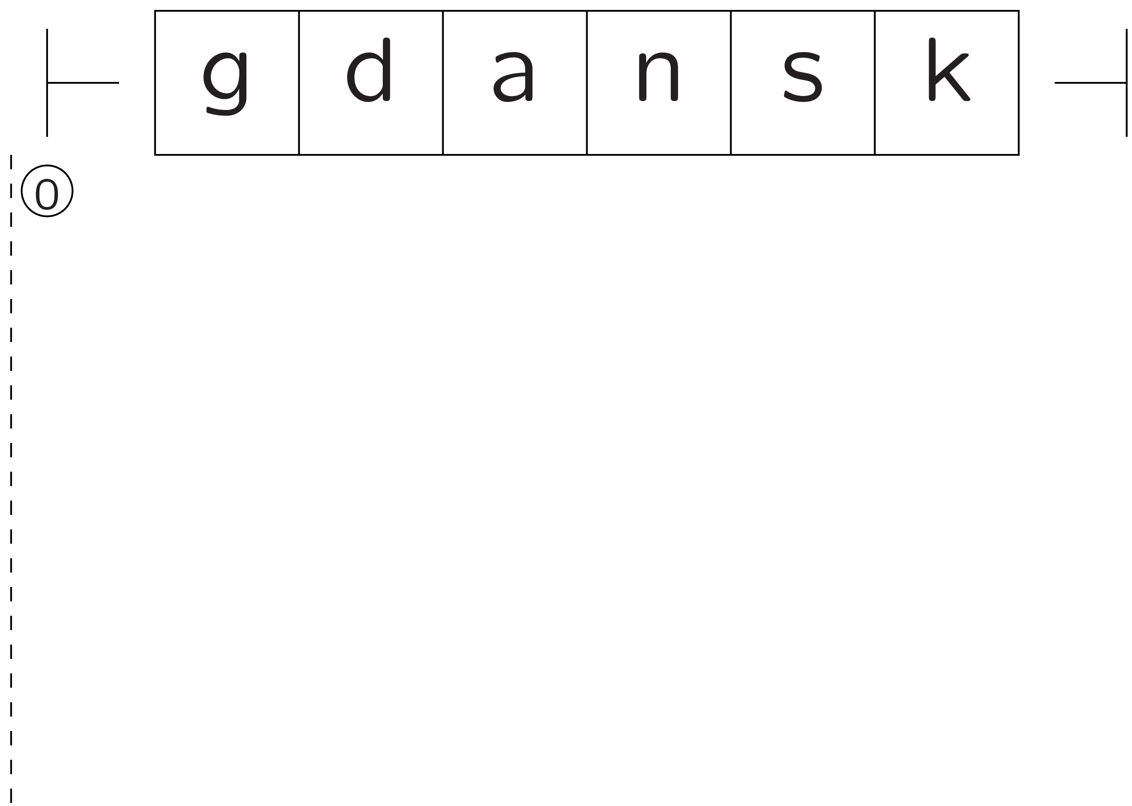
"match" of **frontiers**



$(\{3,5\}, \{1,3,4\})$ $(\{2\}, \{4,8\})$

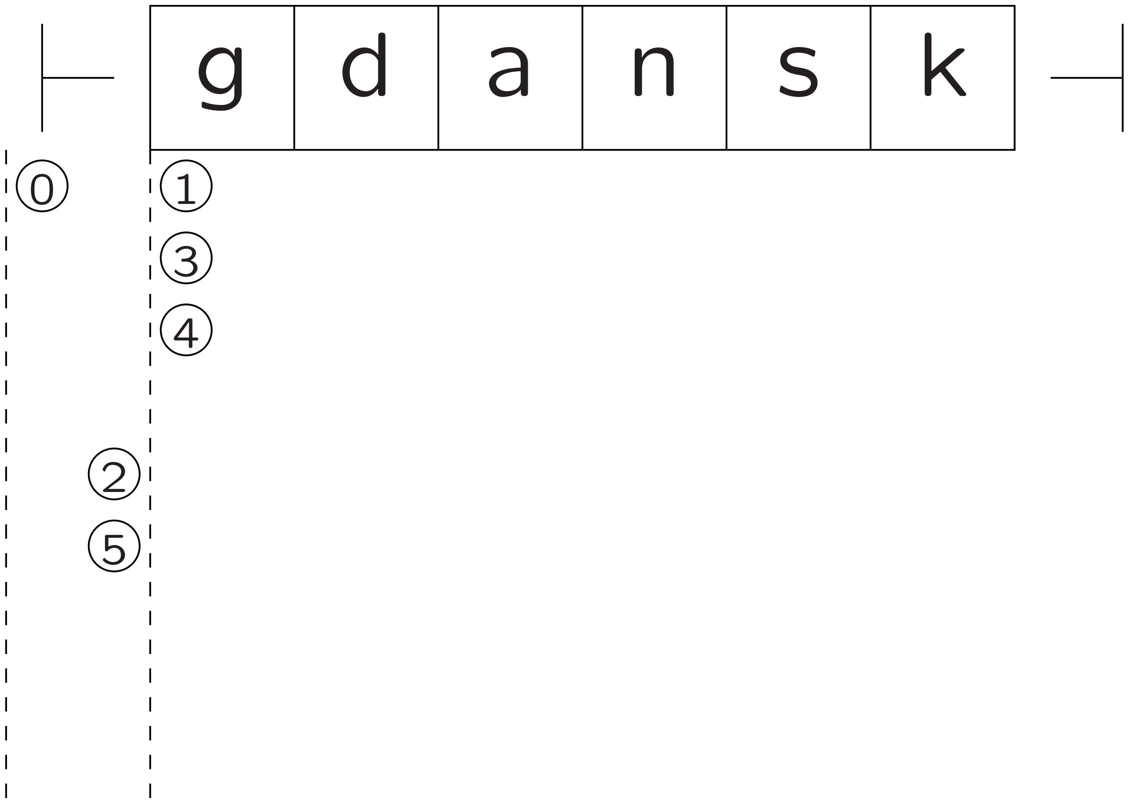
∃ bijection that respects
the trans.function

2NFA accepts $\Leftarrow$ 1NFA accepts

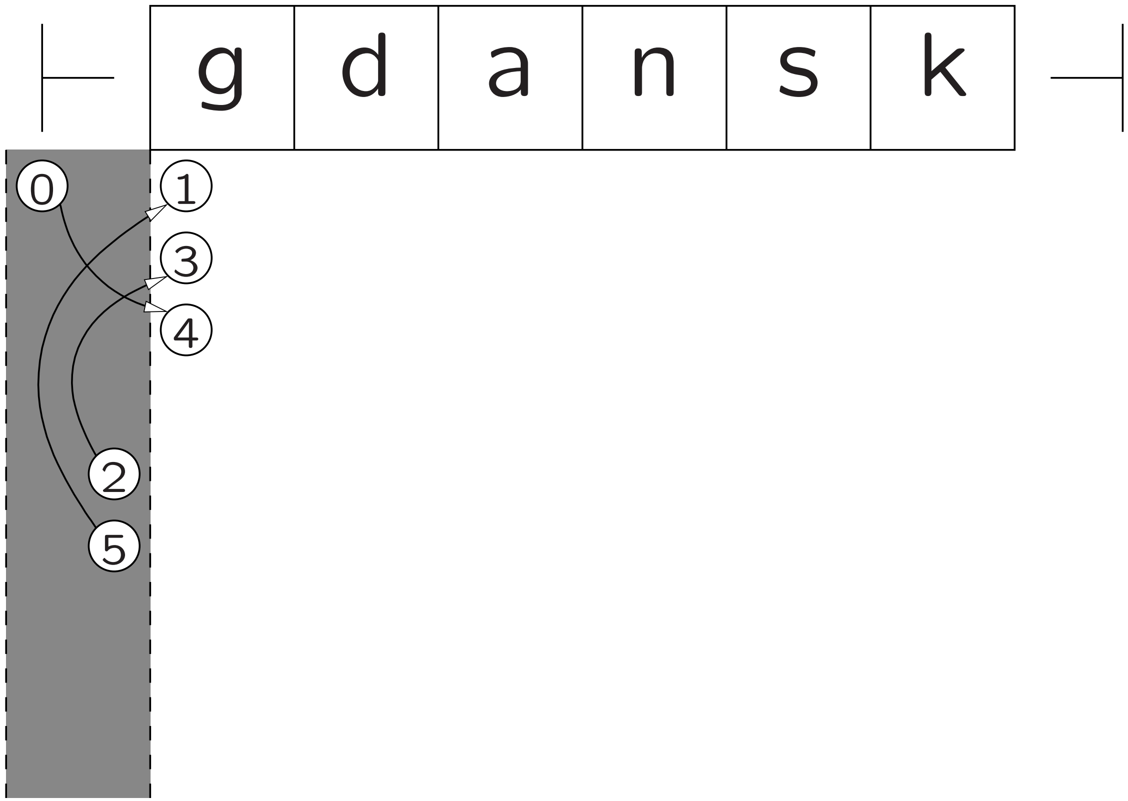


FRONTIER CONSTRUCTION

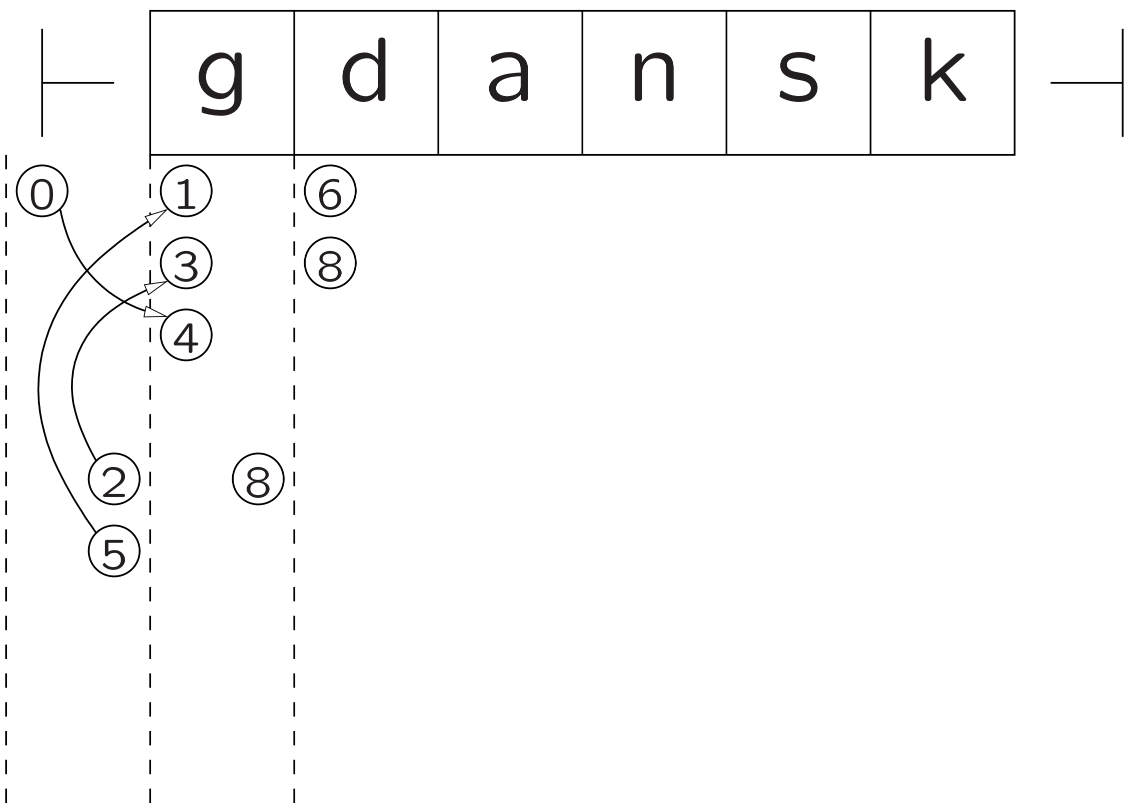
2NFA accepts $\Leftarrow$ 1NFA accepts



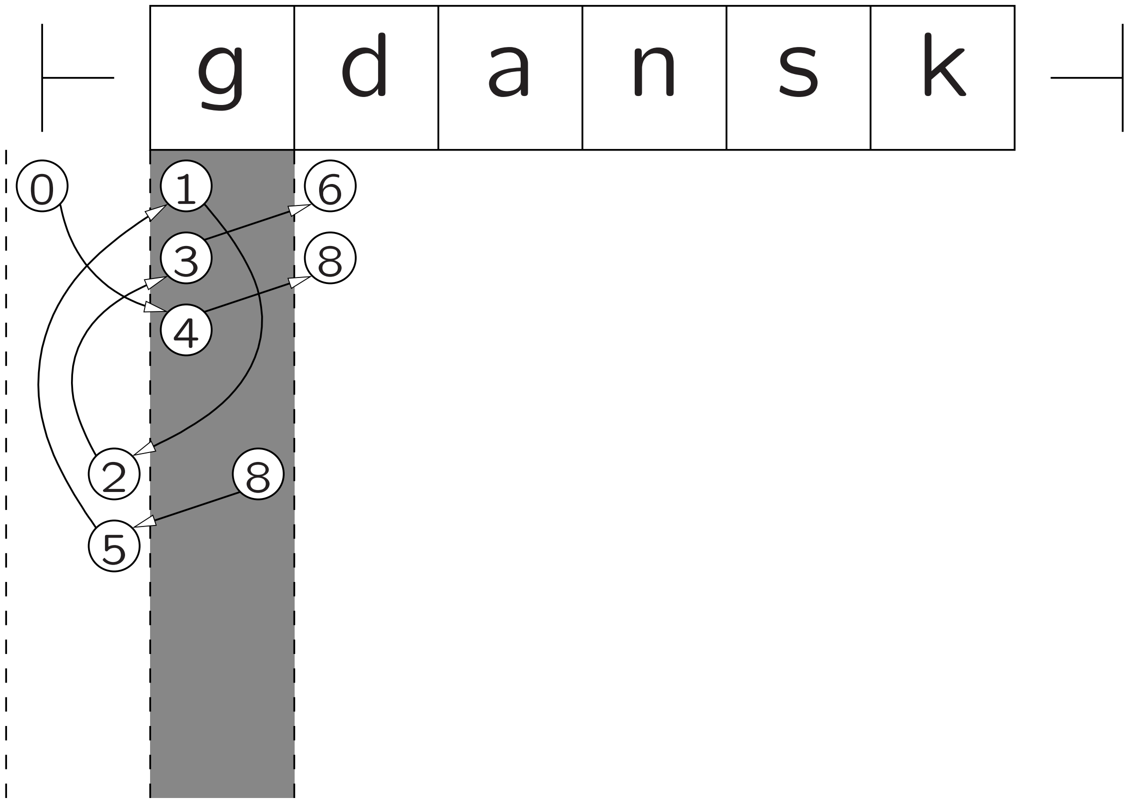
2NFA accepts $\Leftarrow$ 1NFA accepts



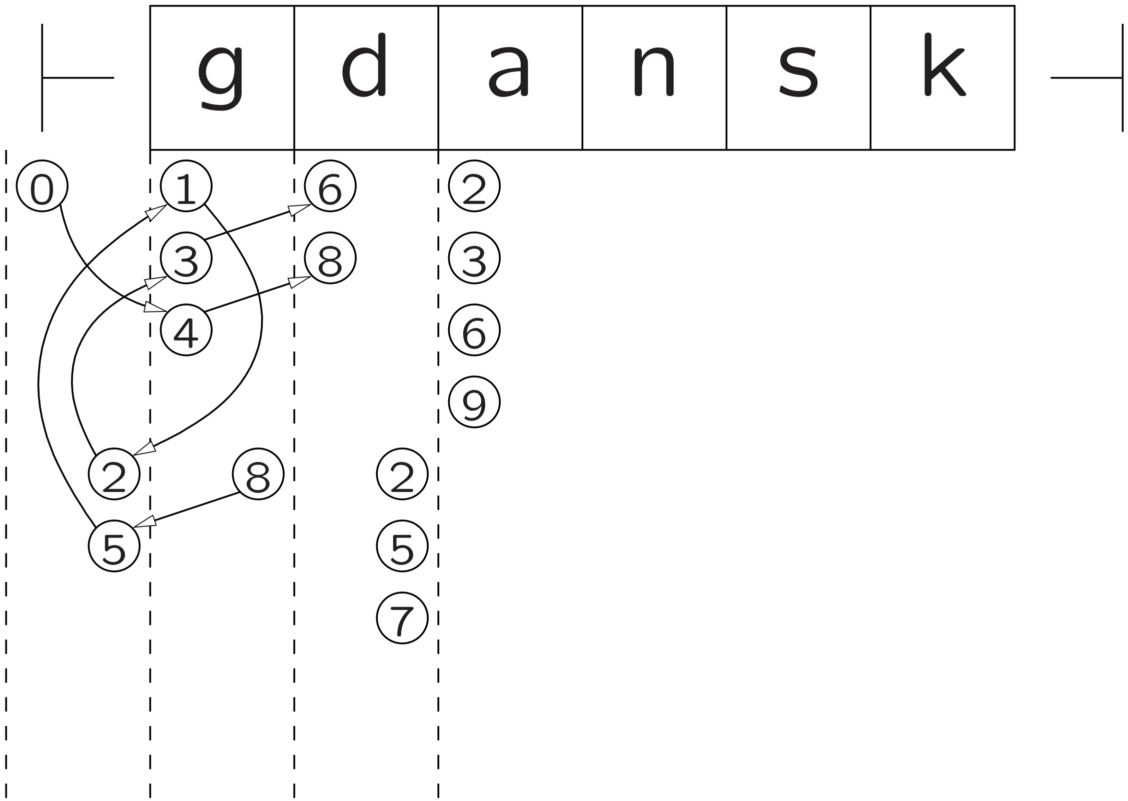
2NFA accepts $\Leftarrow$ 1NFA accepts



2NFA accepts $\Leftarrow$ 1NFA accepts

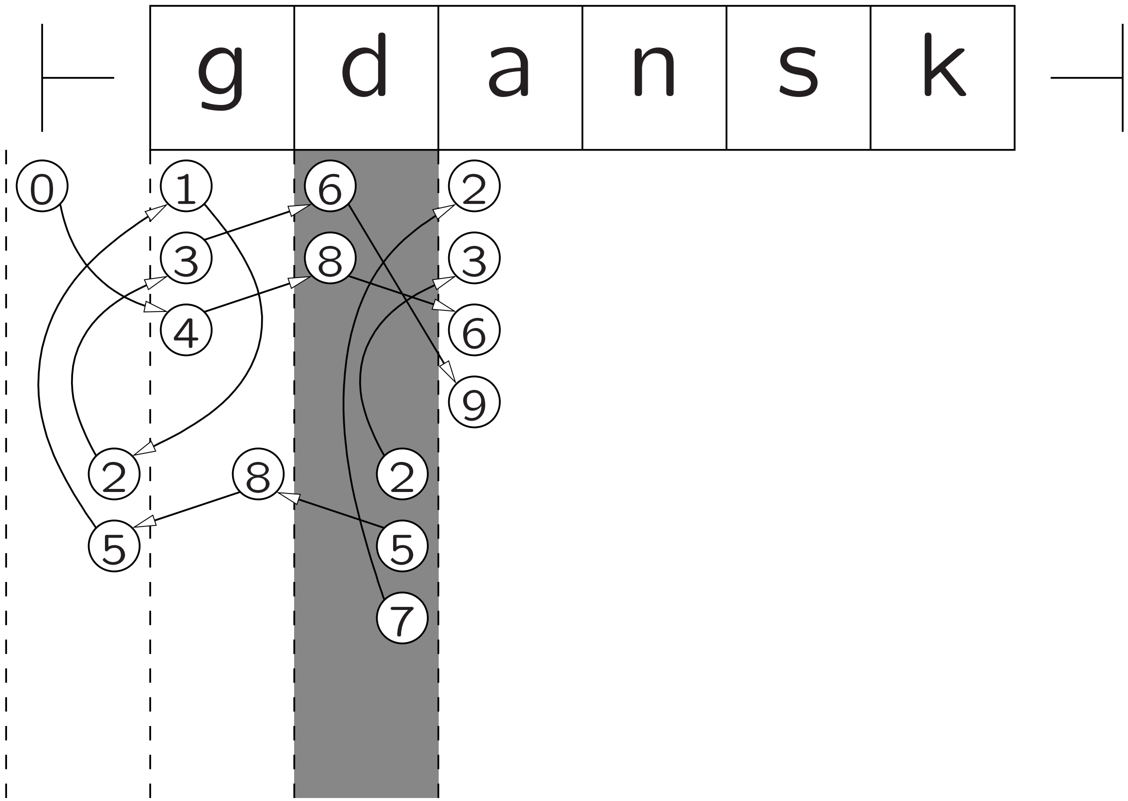


2NFA accepts $\Leftarrow$ 1NFA accepts



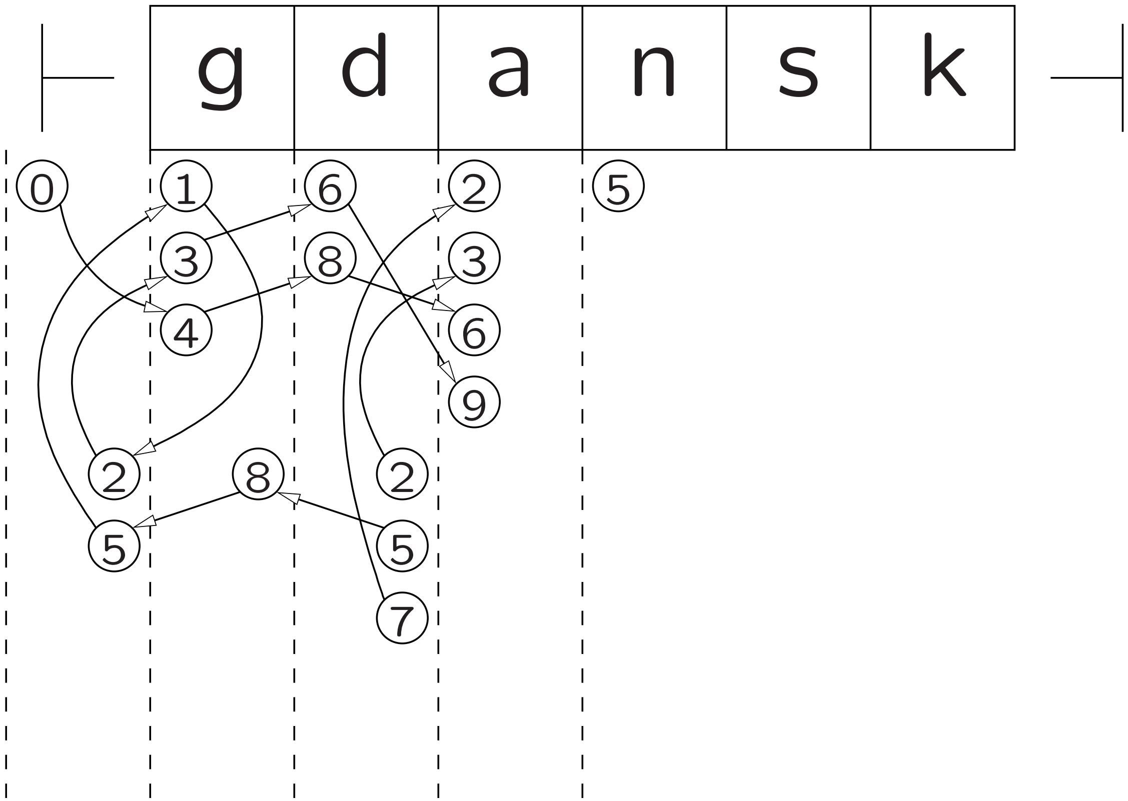
FRONTIER CONSTRUCTION

2NFA accepts $\Leftarrow$ 1NFA accepts

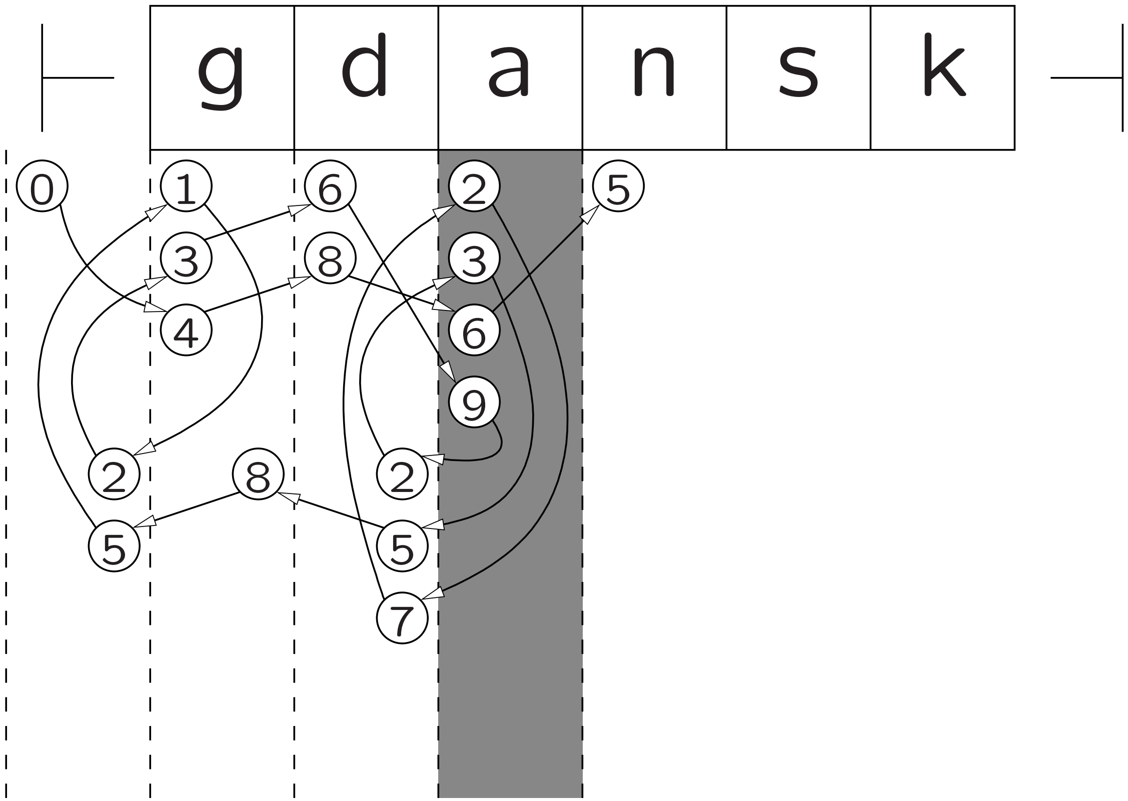


FRONTIER CONSTRUCTION

2NFA accepts $\Leftarrow$ 1NFA accepts

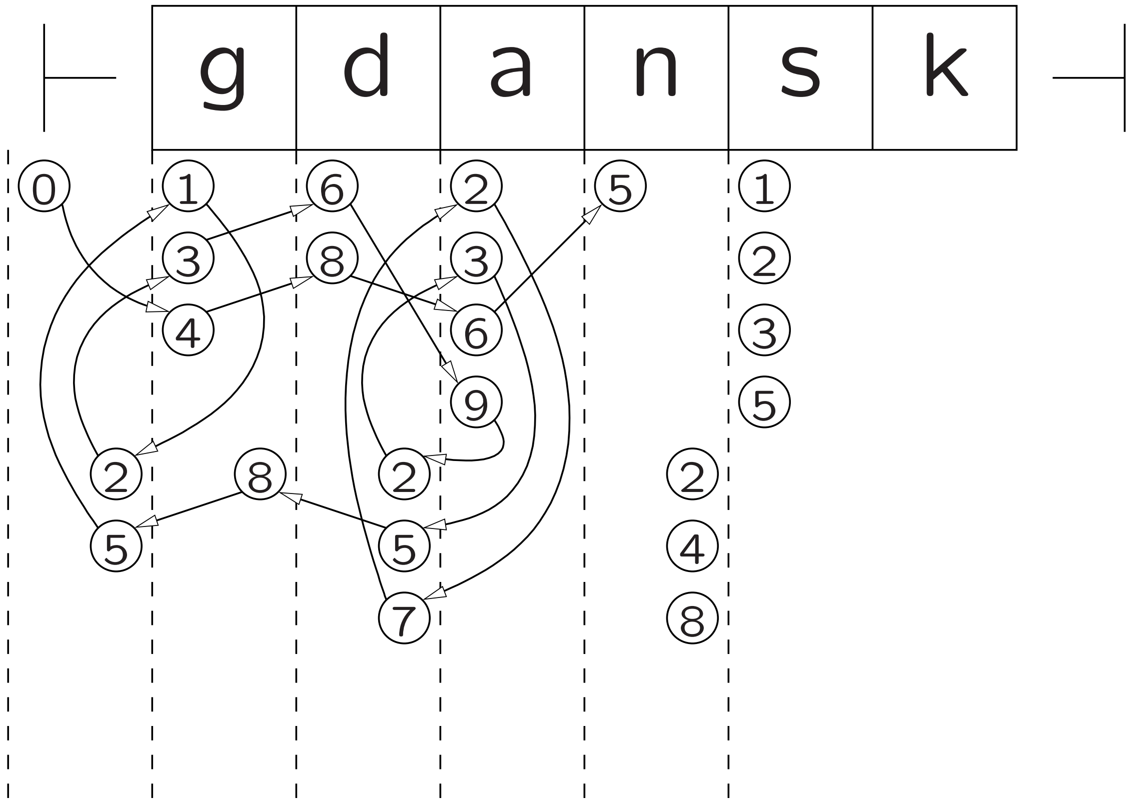


2NFA accepts ⇐ 1NFA accepts

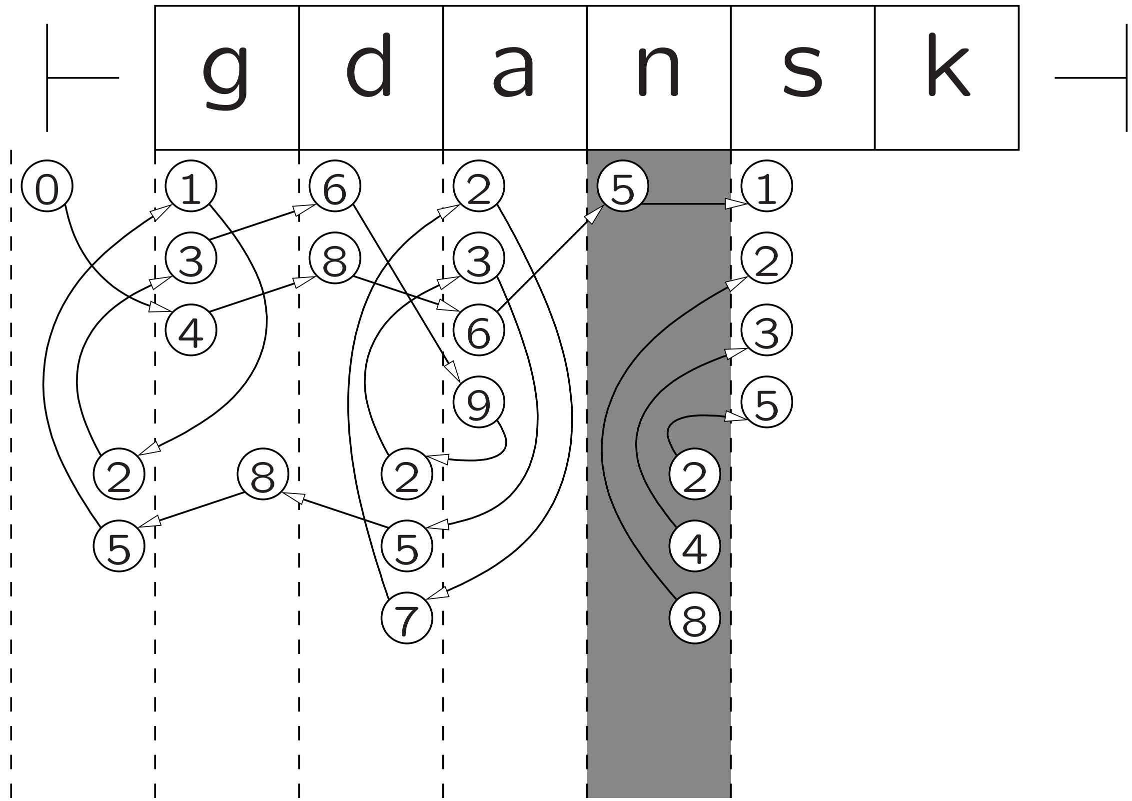


FRONTIER CONSTRUCTION

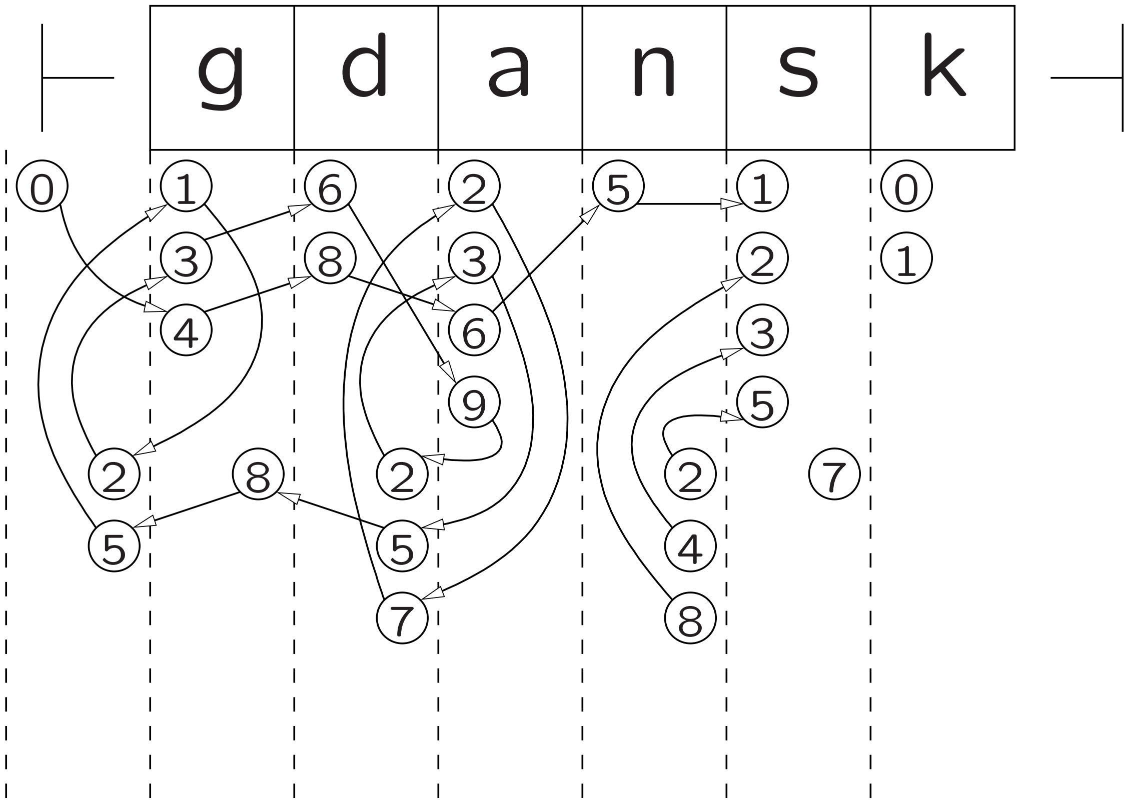
2NFA accepts $\Leftarrow$ 1NFA accepts



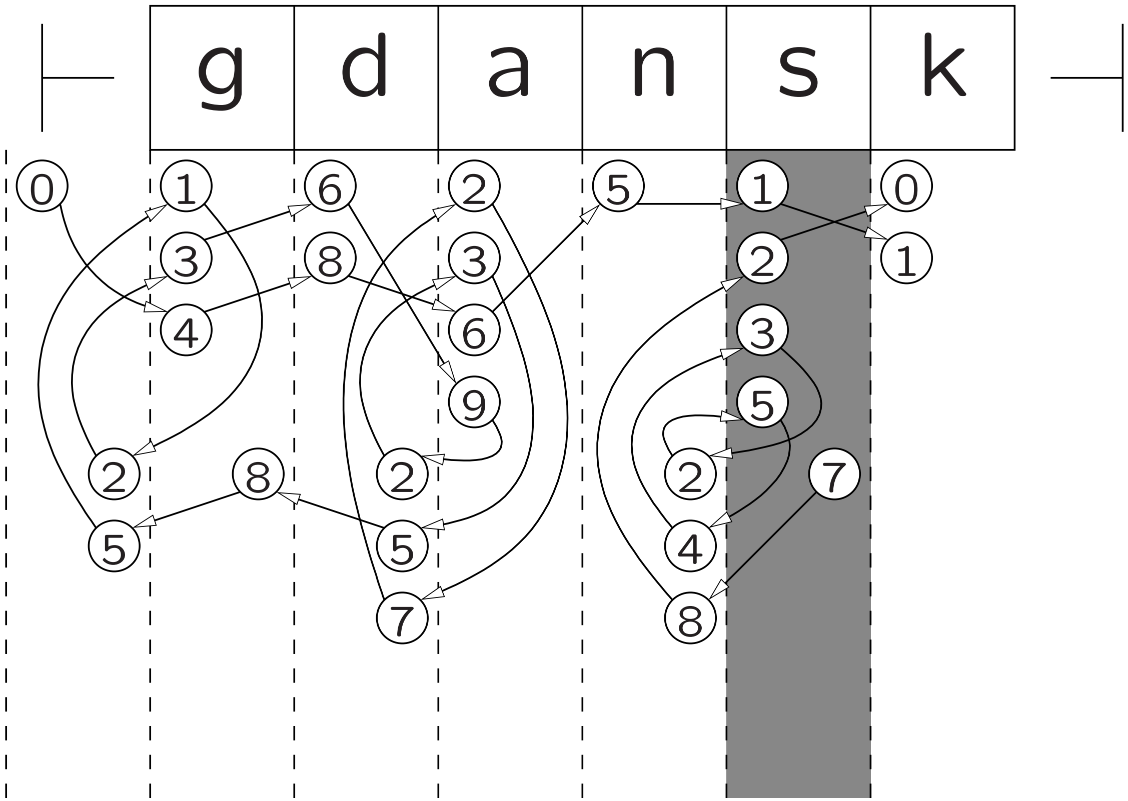
2NFA accepts $\Leftarrow$ 1NFA accepts



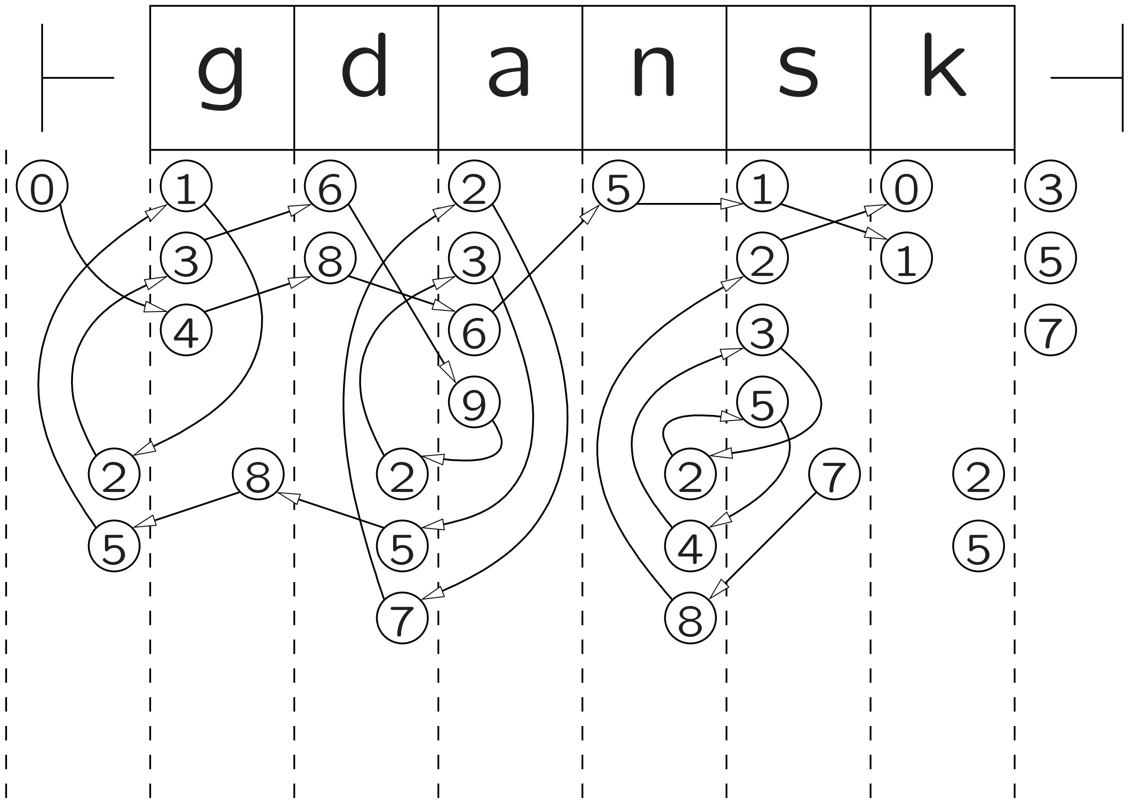
2NFA accepts ⇐ 1NFA accepts



2NFA accepts $\Leftarrow$ 1NFA accepts

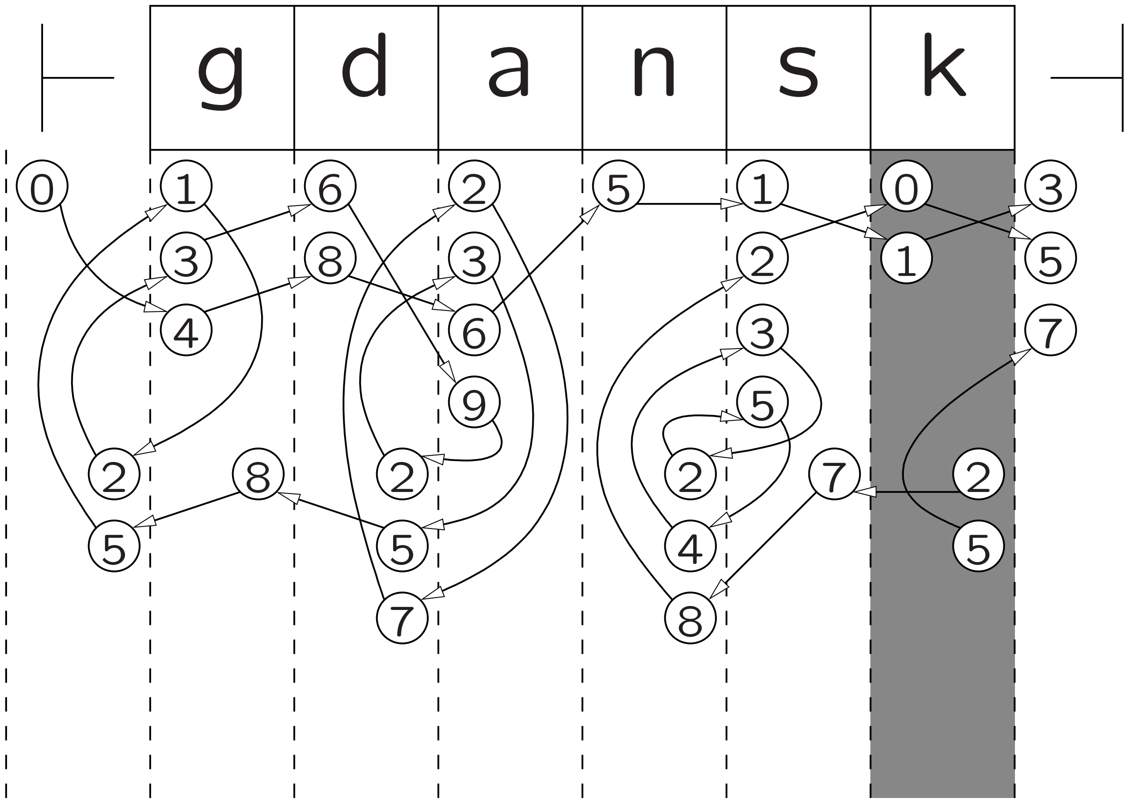


2NFA accepts $\Leftarrow$ 1NFA accepts

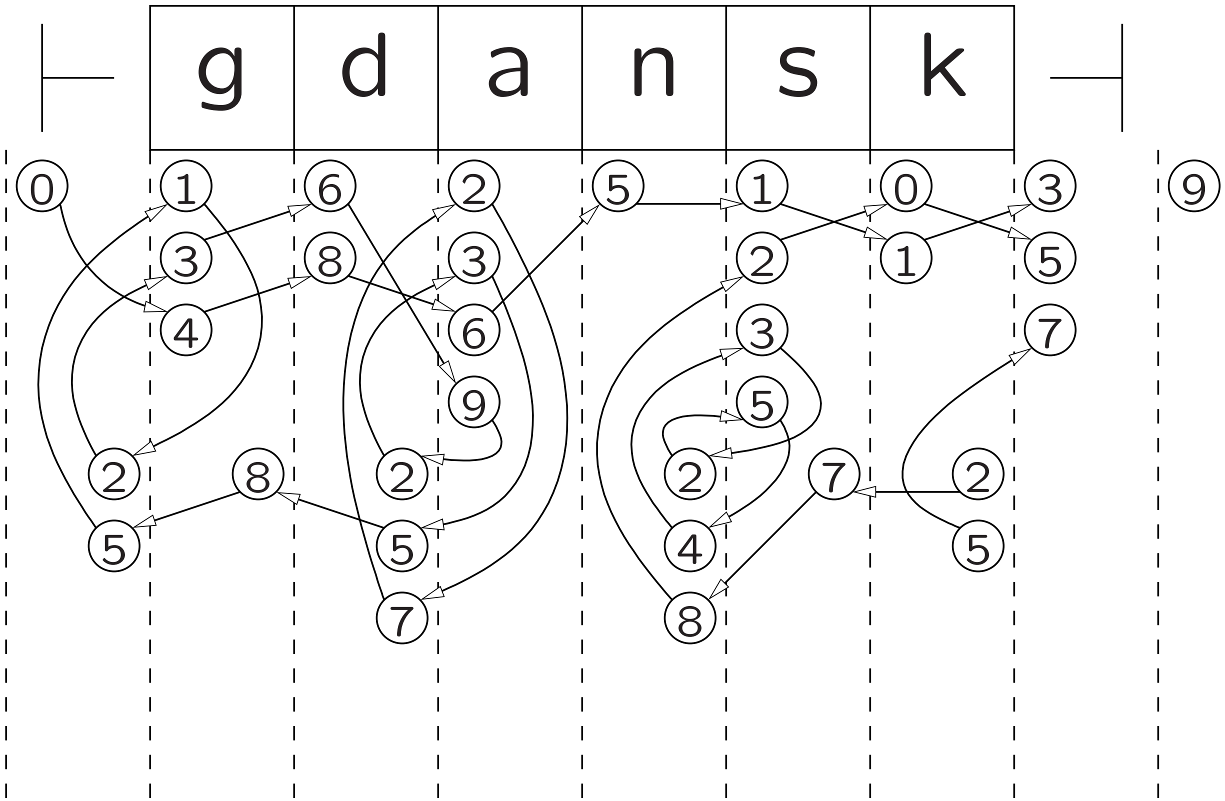


FRONTIER CONSTRUCTION

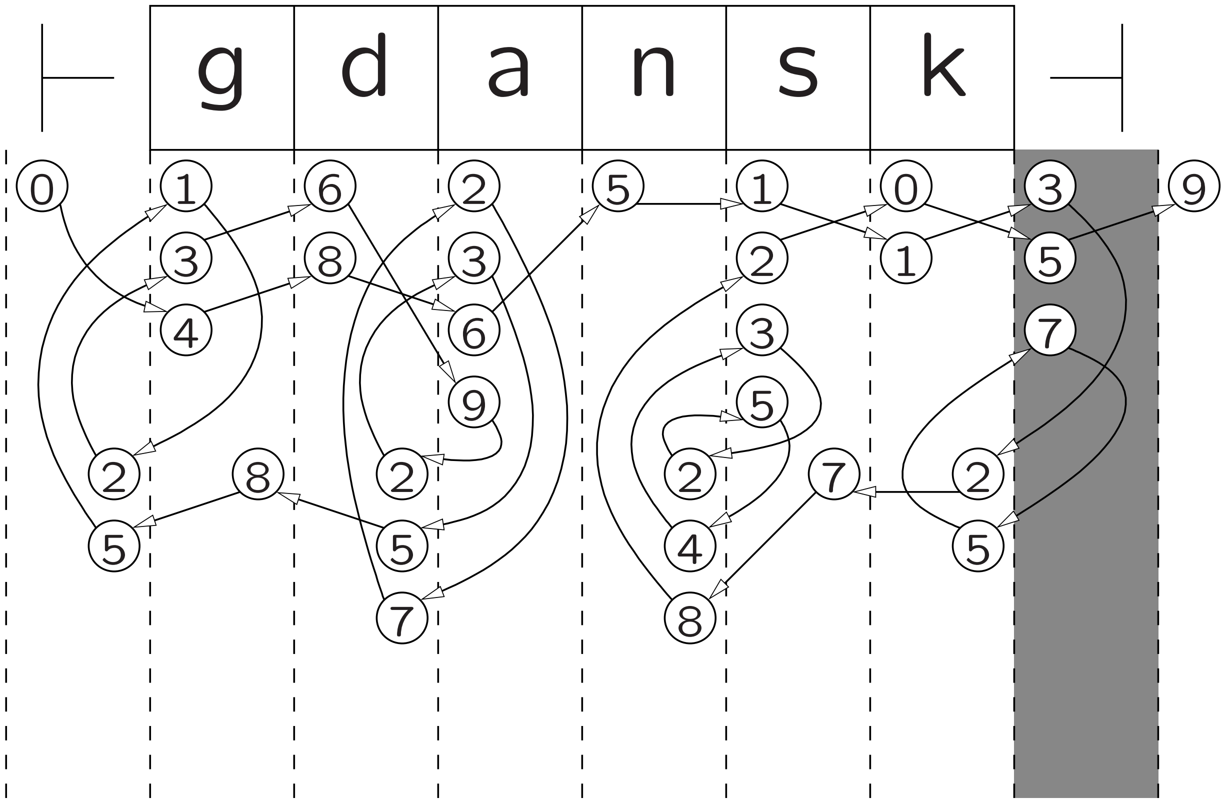
2NFA accepts $\Leftarrow$ 1NFA accepts



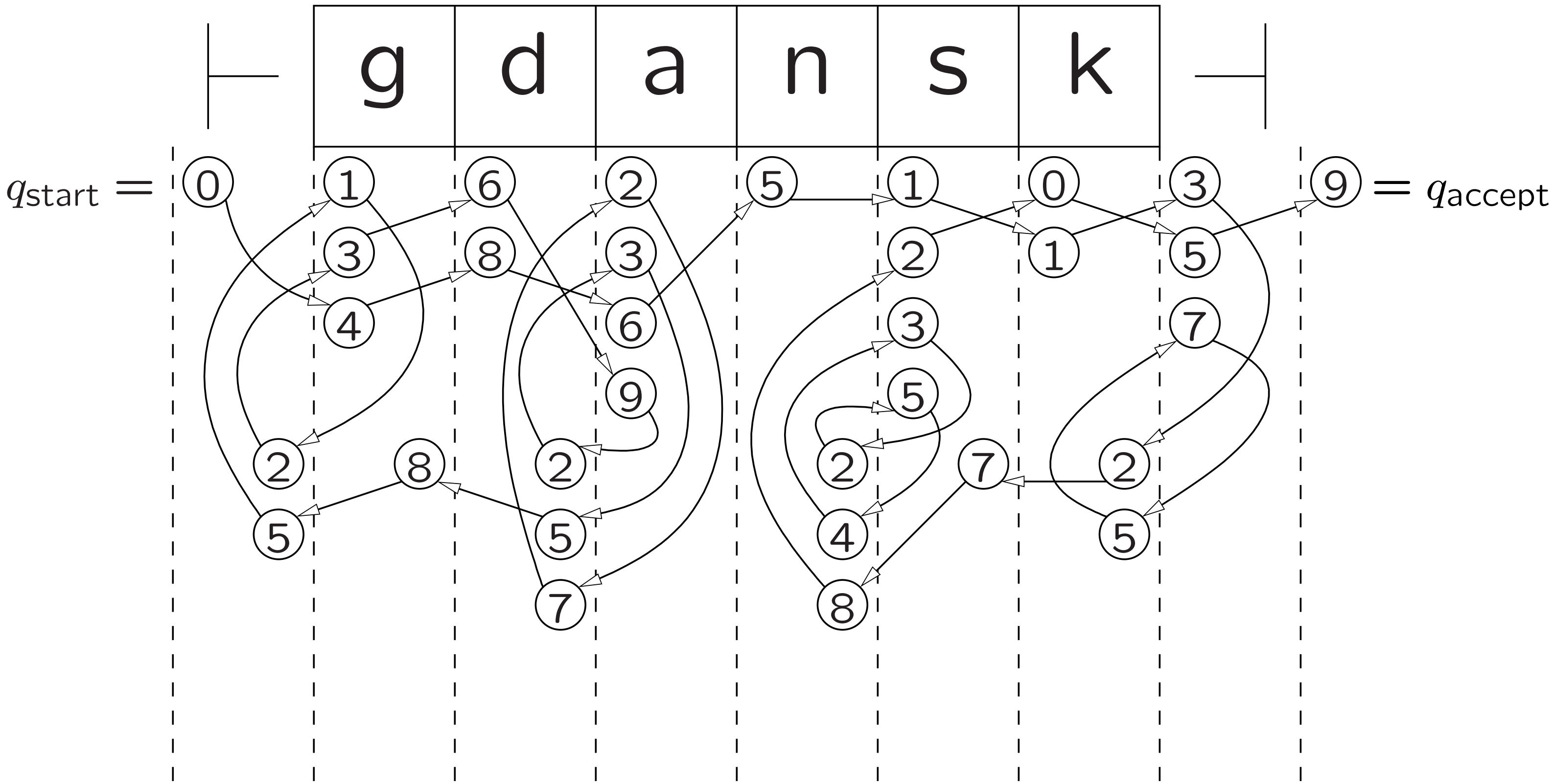
2NFA accepts $\Leftarrow$ 1NFA accepts



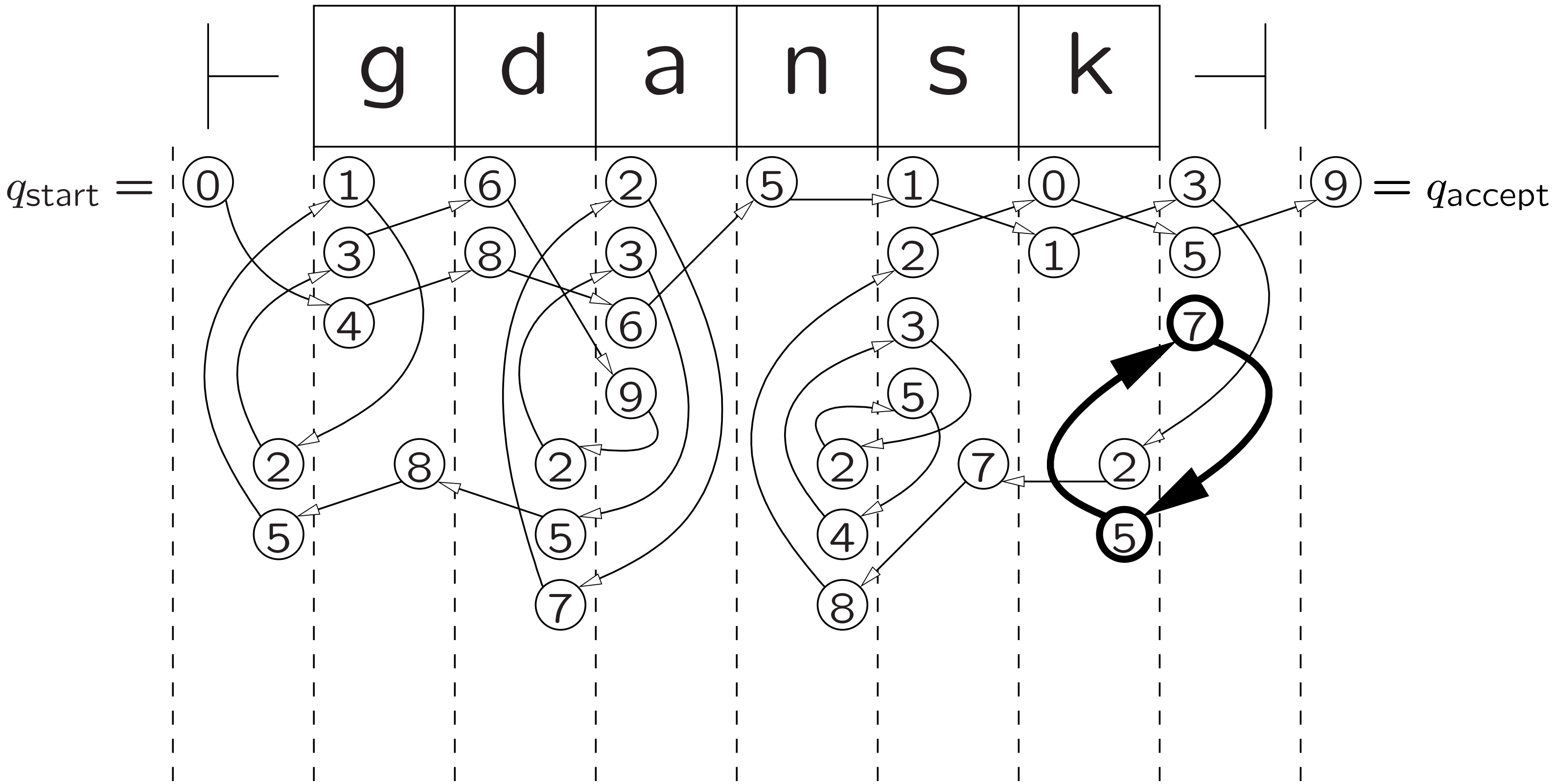
2NFA accepts ⇐ 1NFA accepts



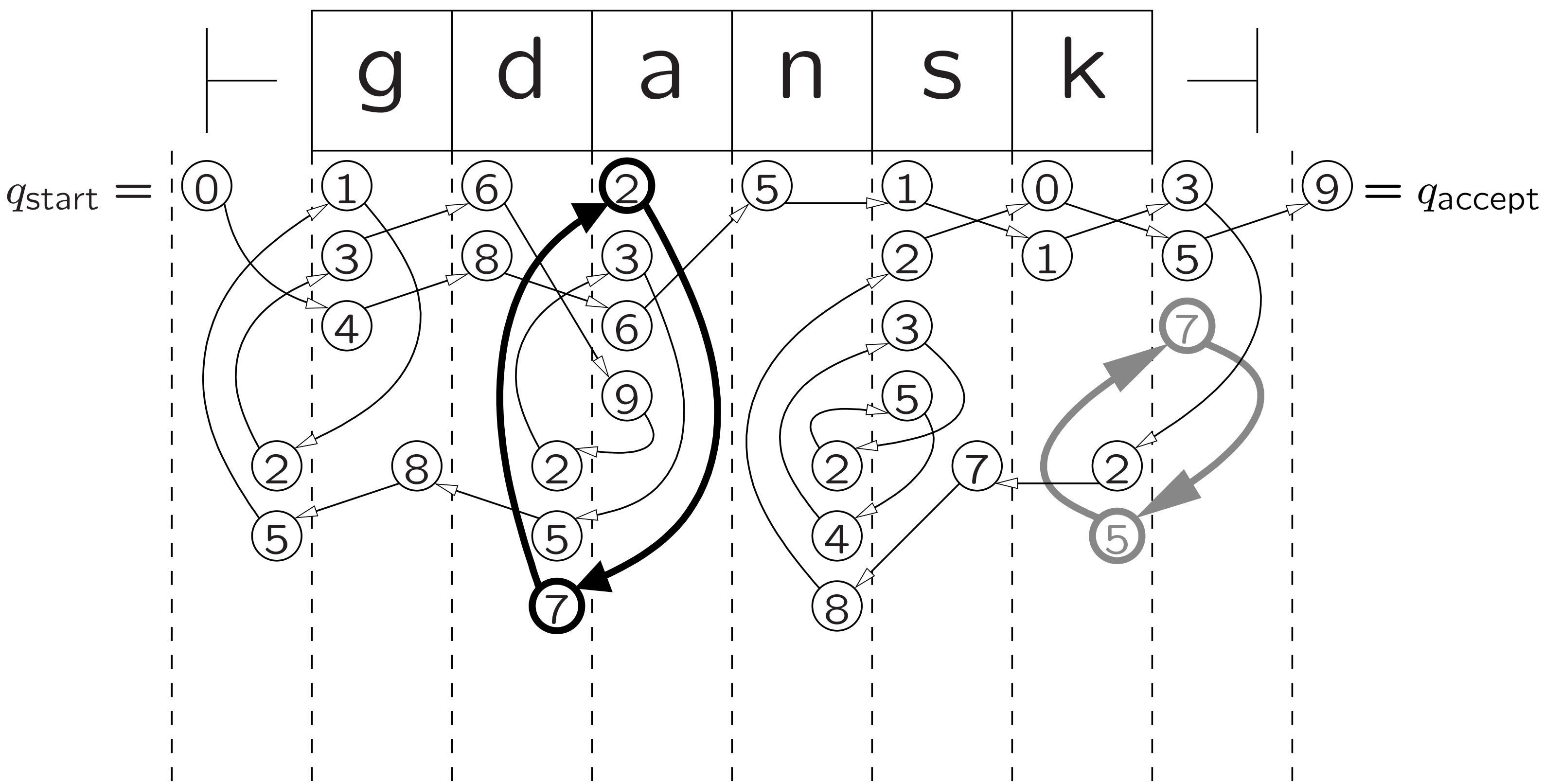
2NFA accepts $\Leftarrow$ 1NFA accepts



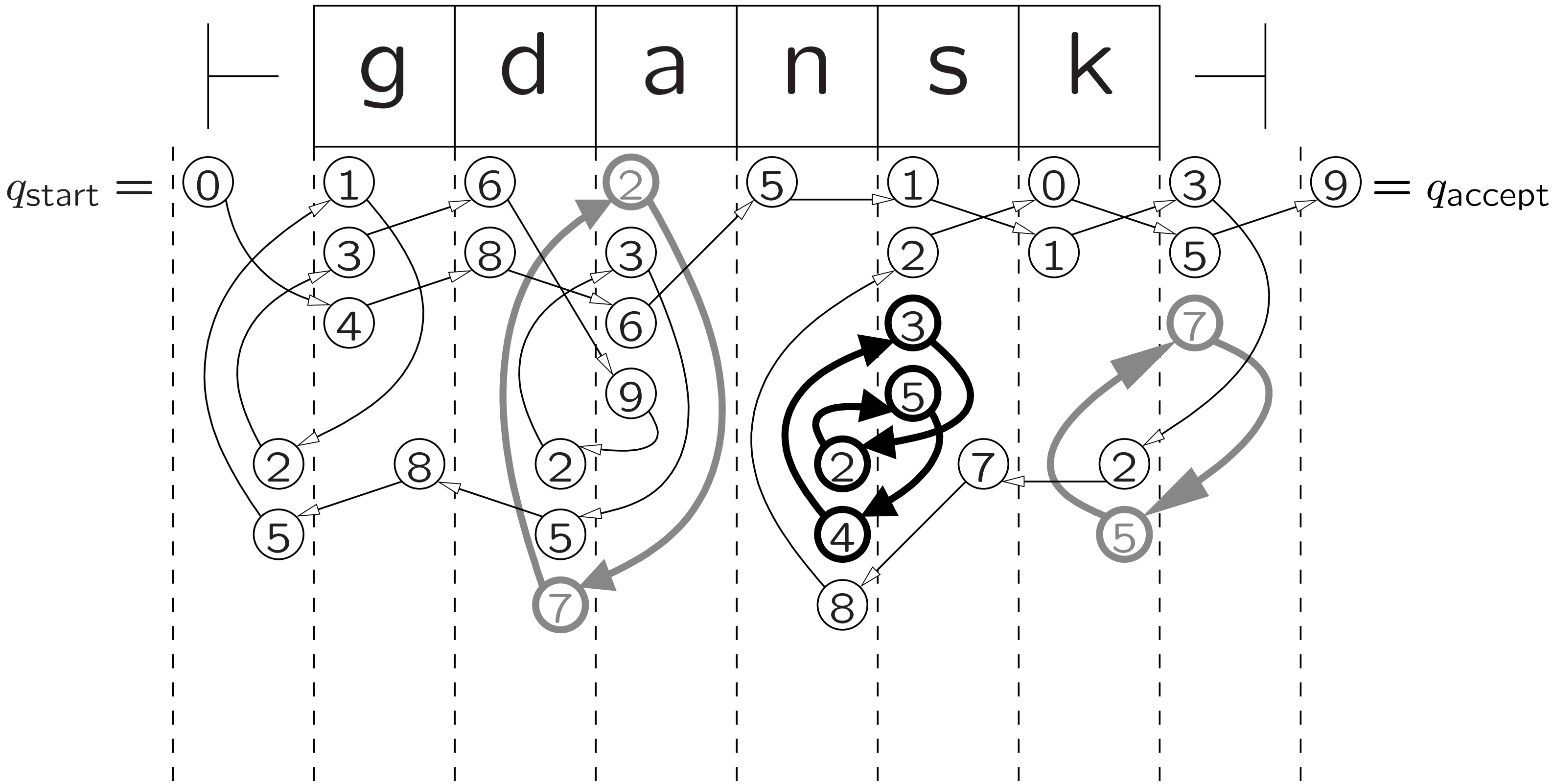
2NFA accepts $\Leftarrow$ 1NFA accepts



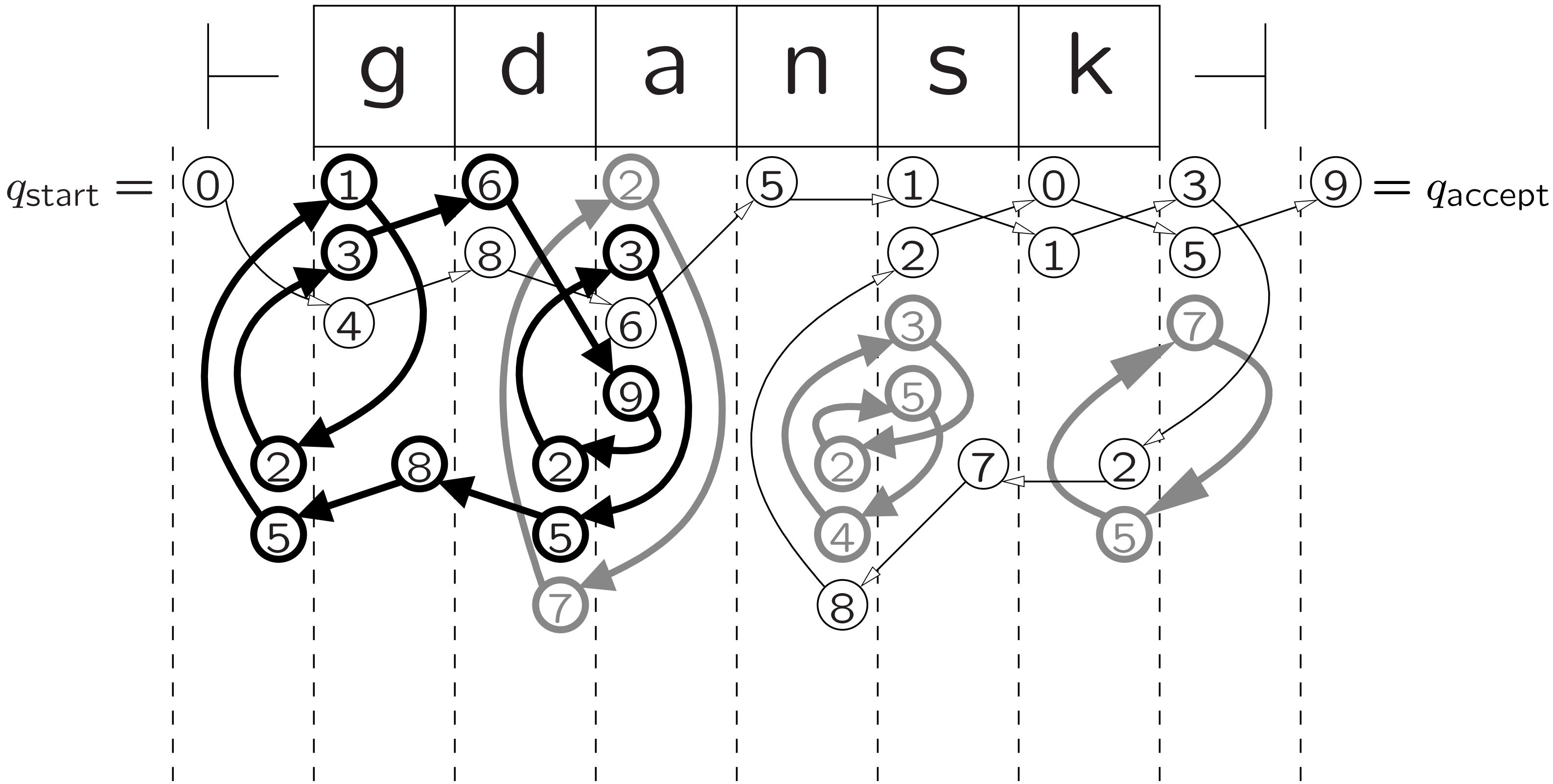
2NFA accepts $\Leftarrow$ 1NFA accepts



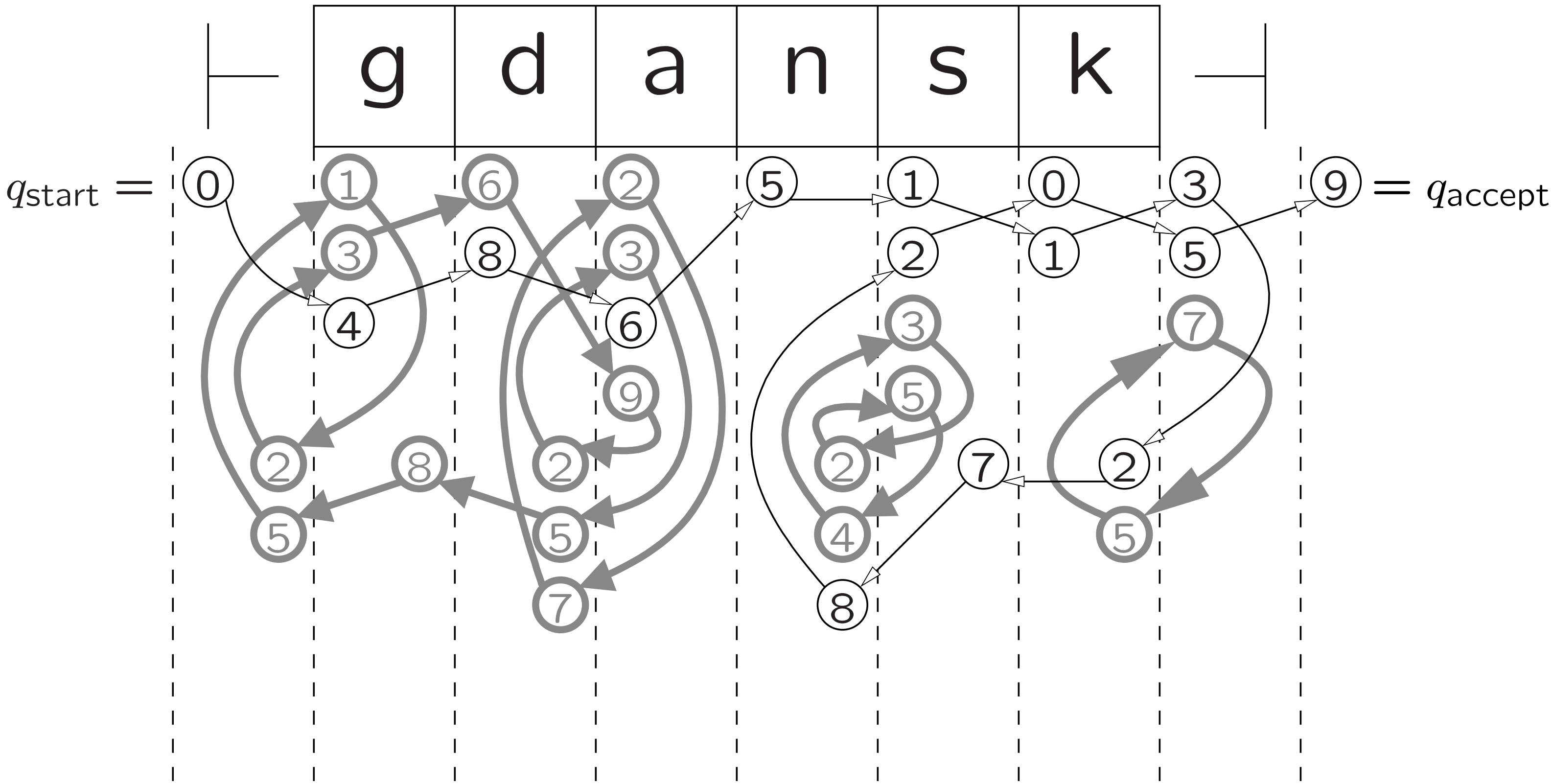
2NFA accepts $\Leftarrow$ 1NFA accepts



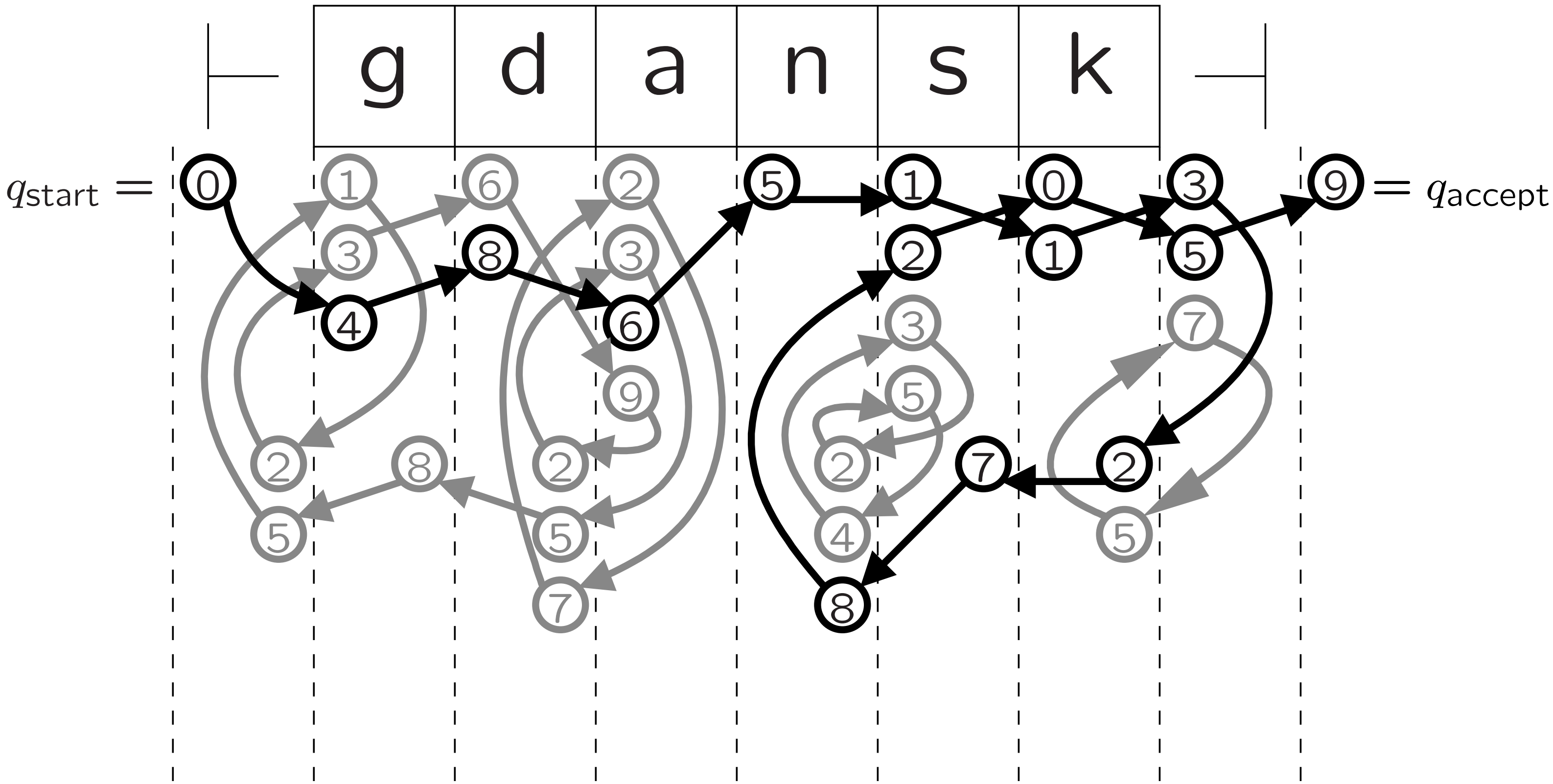
2NFA accepts $\Leftarrow$ 1NFA accepts



2NFA accepts $\Leftarrow$ 1NFA accepts



2NFA accepts $\Leftarrow$ 1NFA accepts



FRONTIER CONSTRUCTION

SIMULATING 1NFA:

states	$=$	all **frontiers** of the 2NFA
start state	$=$	$\big(\emptyset, \{q_{\text{start}}\}\big)$
accept state	$=$	$\big(\emptyset, \{q_{\text{accept}}\}\big)$
$\delta\big(F, \text{a}\big)$	$=$	{all **frontiers** that match with F under a}

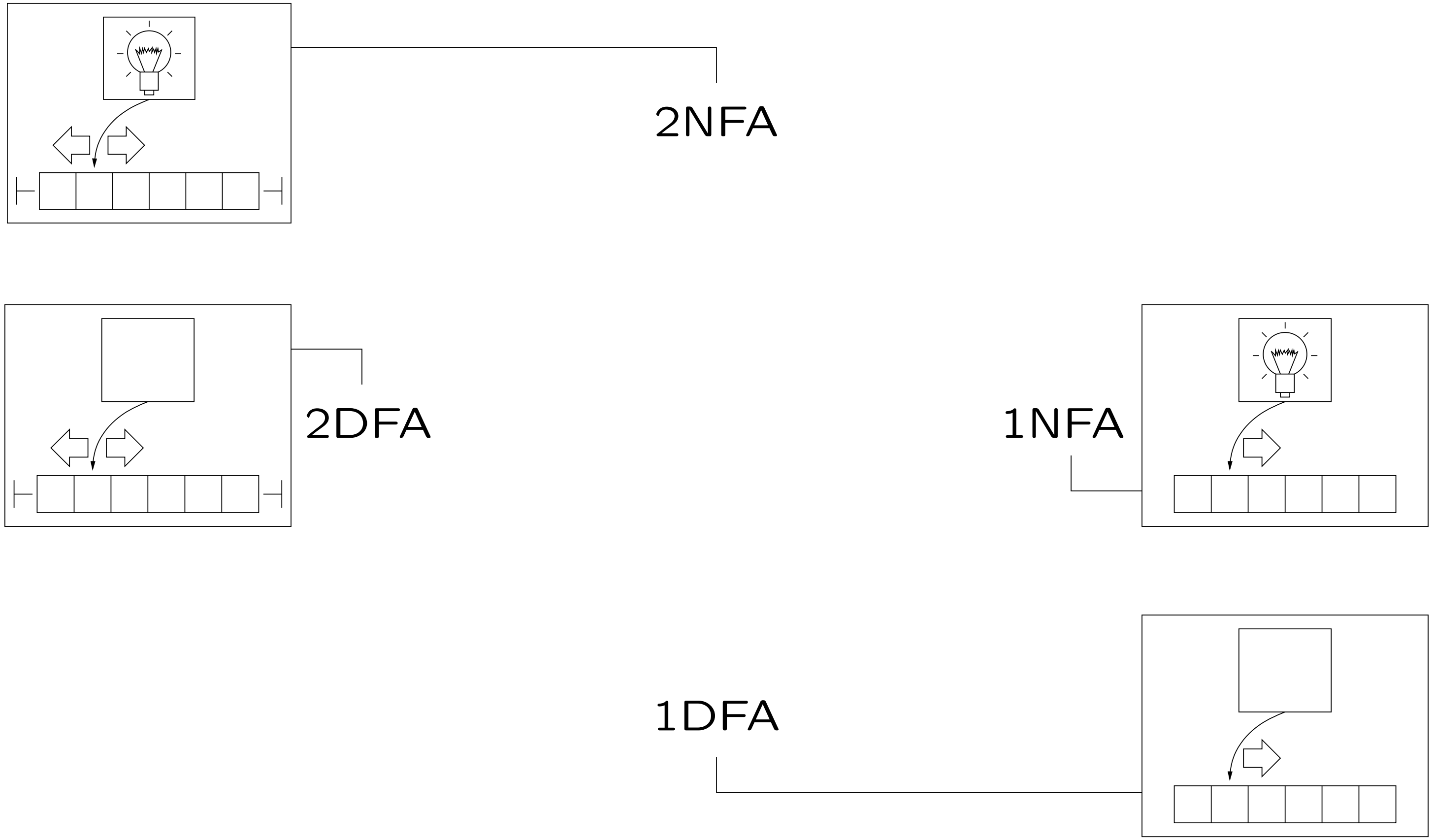
TOTAL SIZE: exactly $\binom{2n}{n+1}$

- every n-state 2NFA has an equivalent 1NFA with $\leq \binom{2n}{n+1}$ states

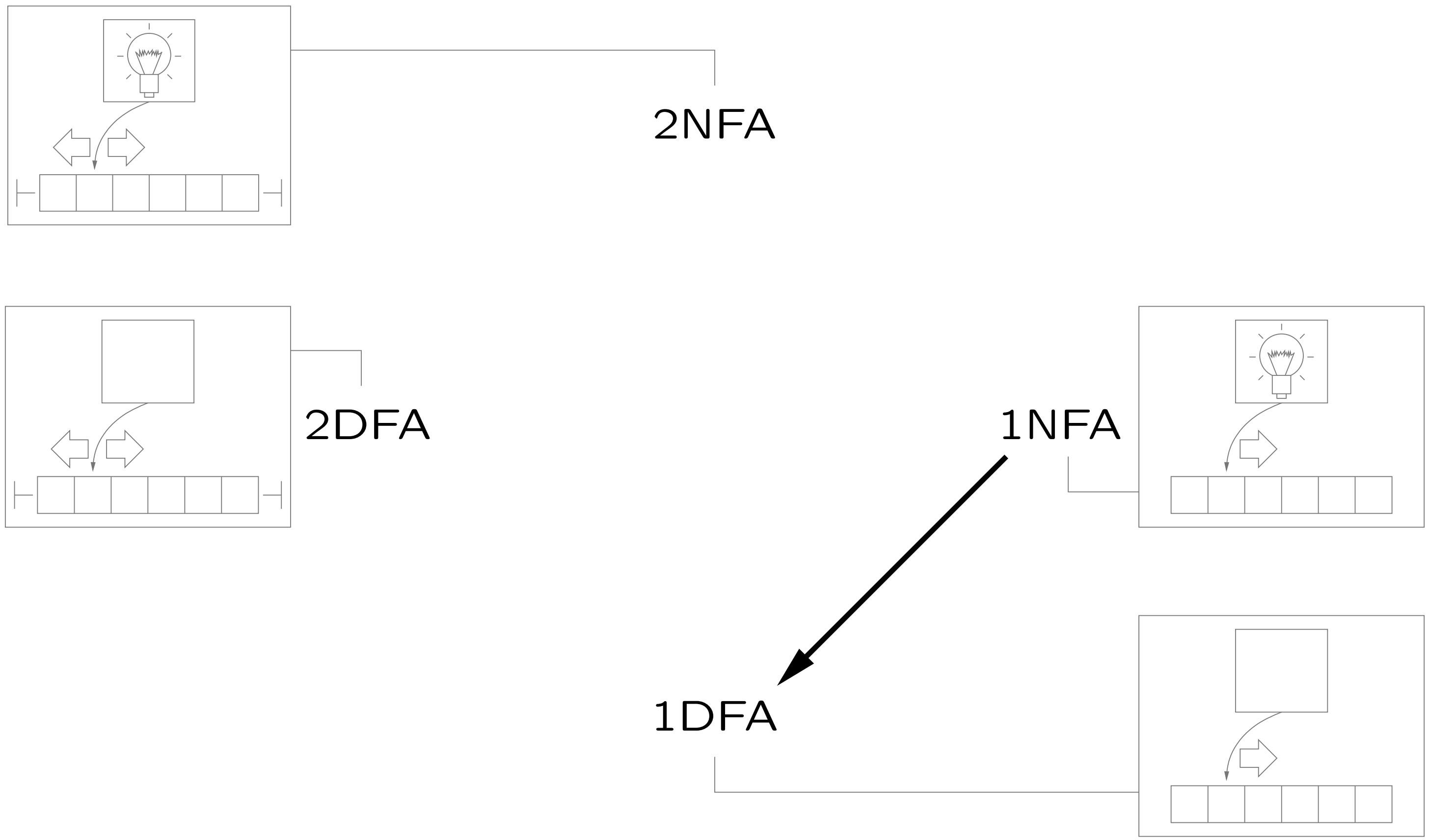
- some n-state 2NFA has no equivalent 1NFA with $< \binom{2n}{n+1}$ states

- hence, the trade-off from 2NFAs to 1NFAs is exactly $\binom{2n}{n+1}$

the big picture

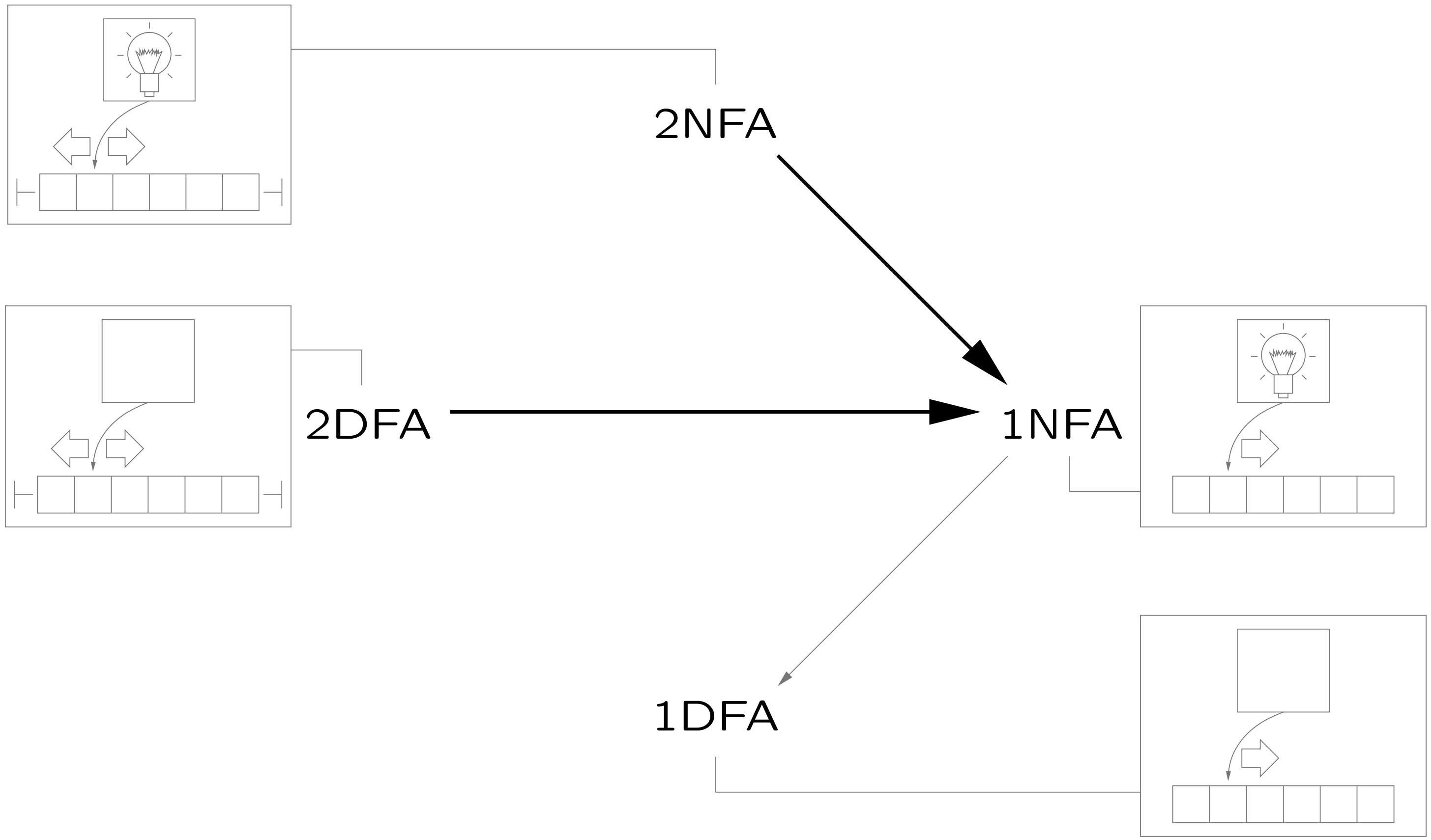


the big picture



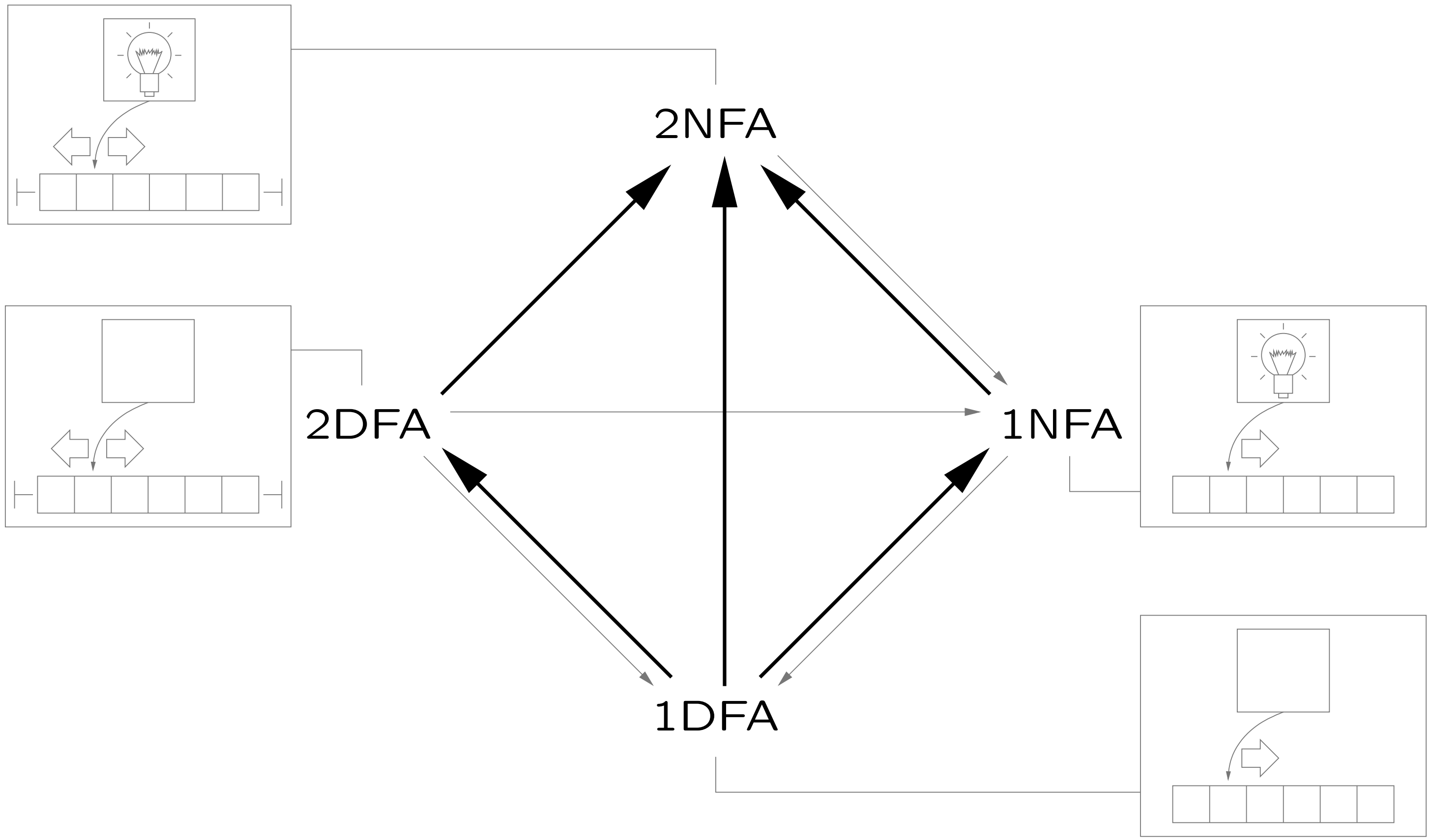
$2^n - 1$

the big picture

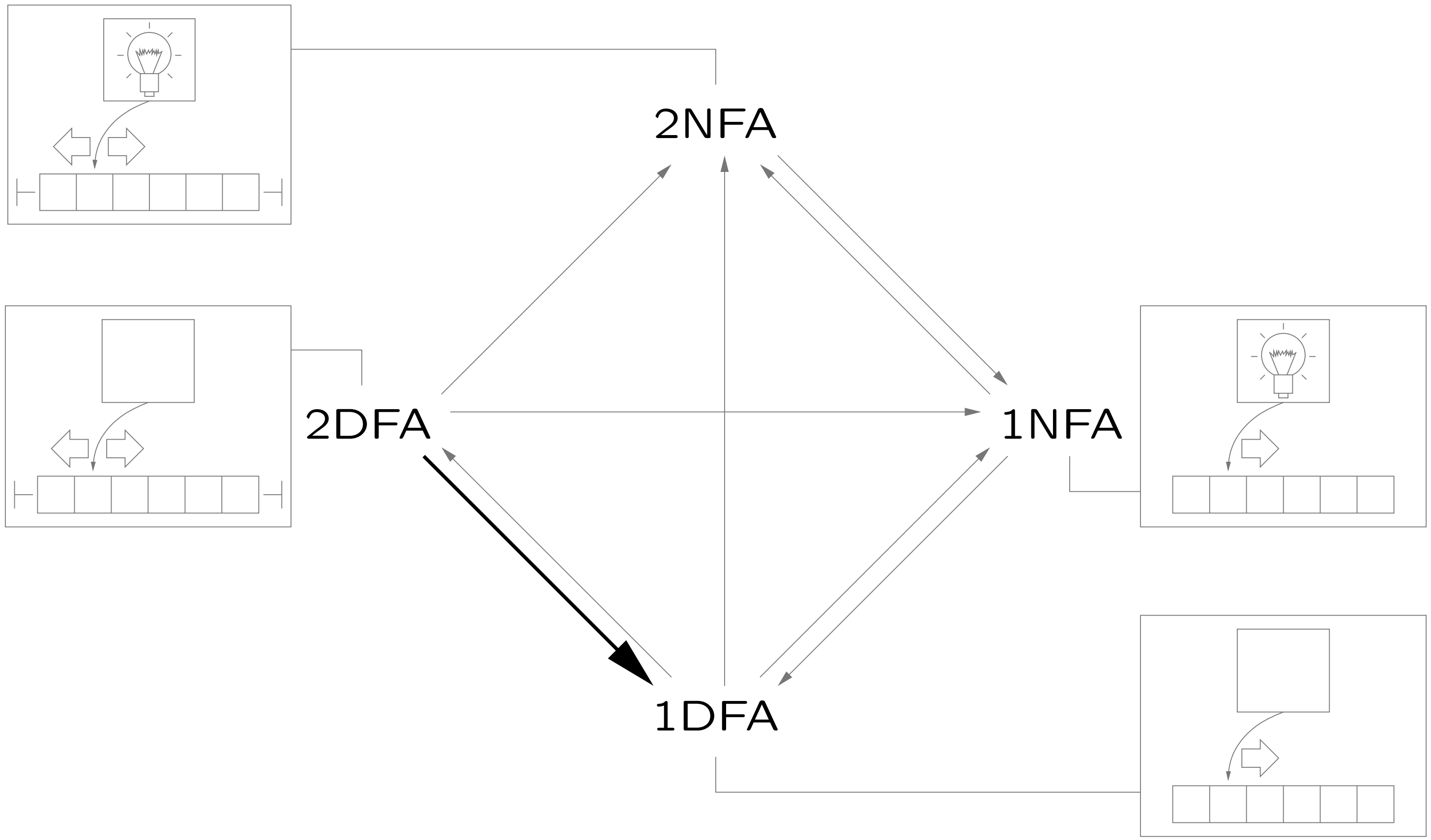


$2^n - 1, \binom{2n}{n+1}$

the big picture

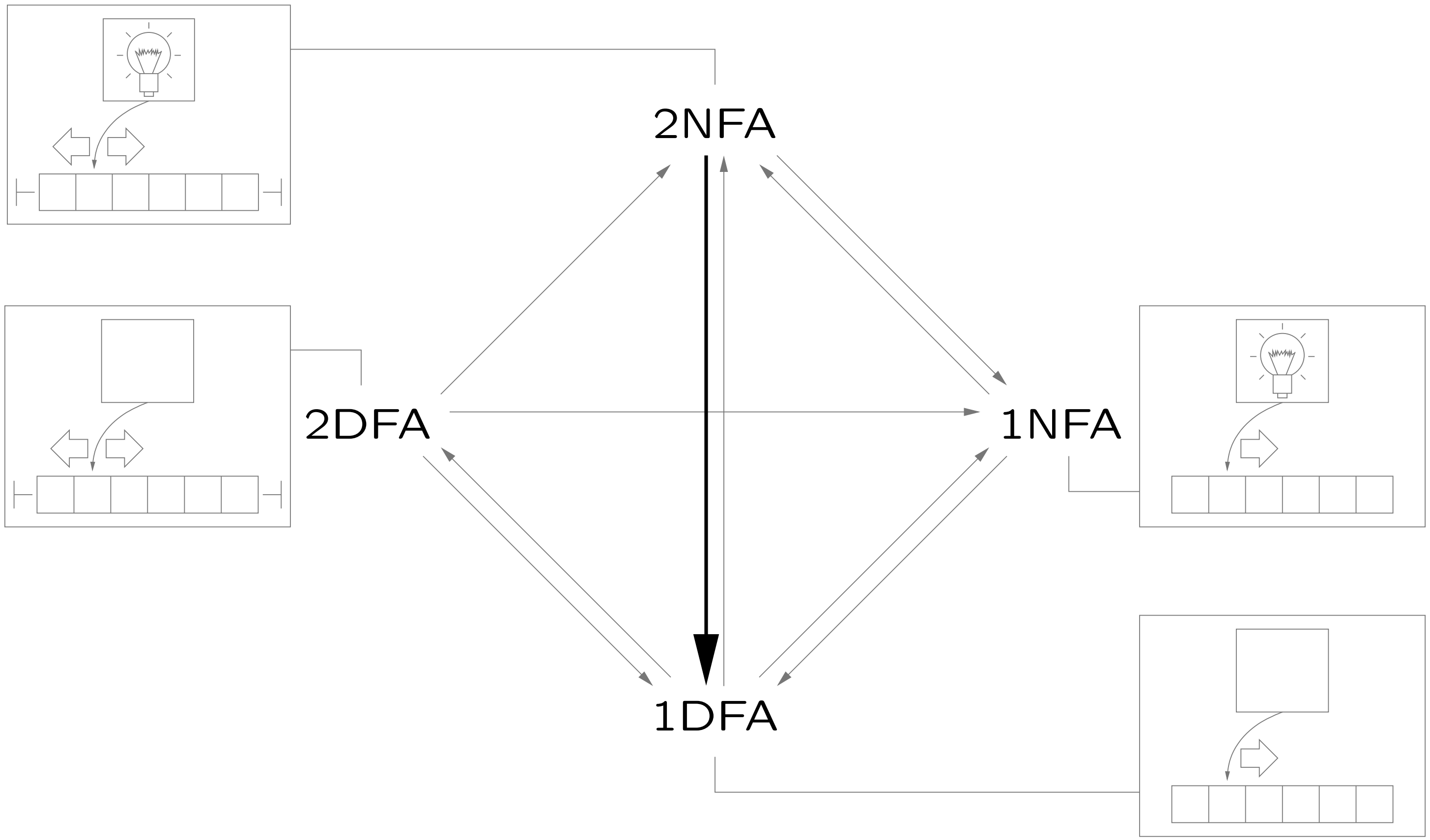


$2^n - 1$, $\binom{2n}{n+1}$, n



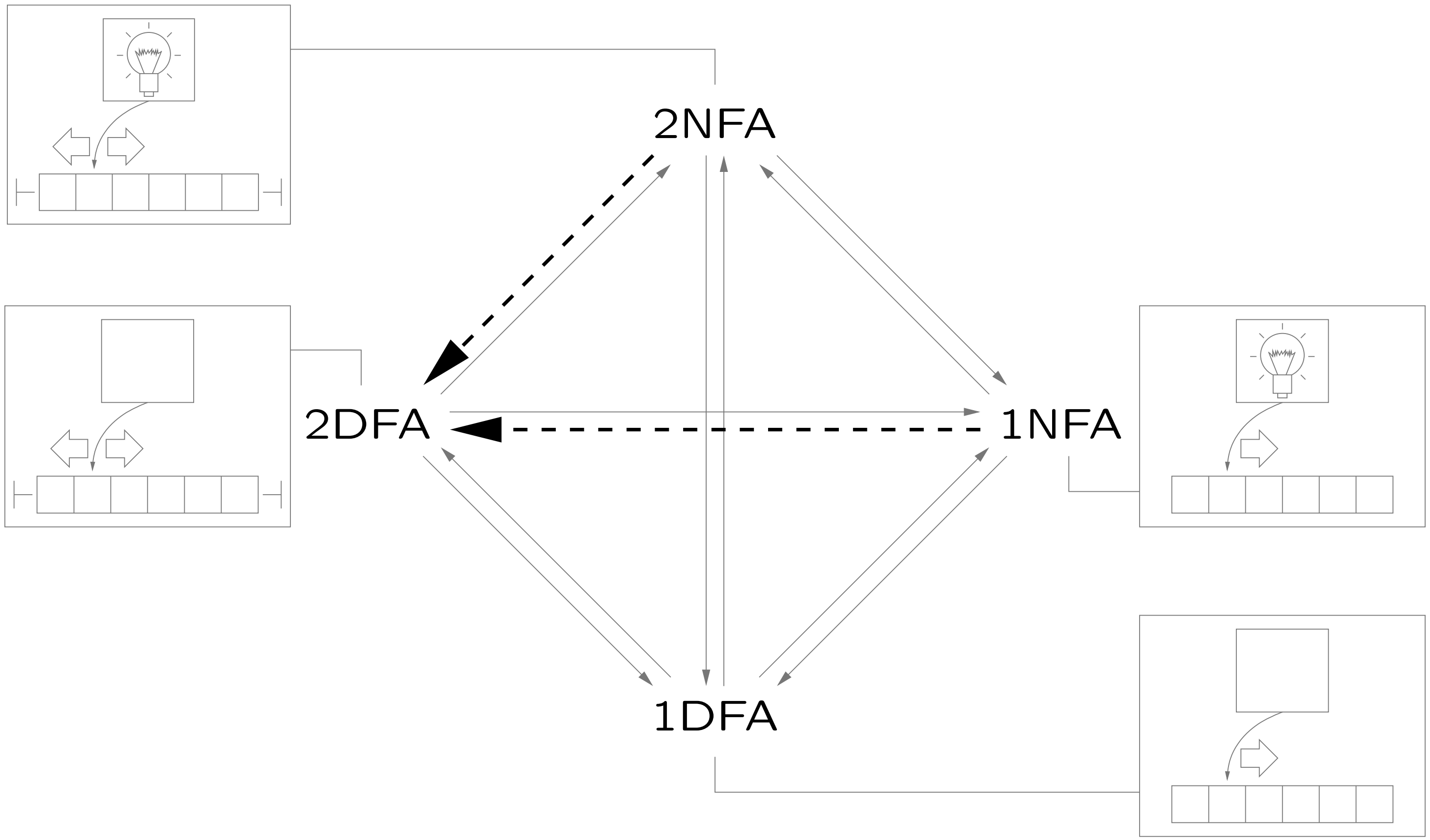
$2^n - 1$, $\binom{2n}{n+1}$, n, $n(n^n - (n-1)^n)$

the big picture



$2^n - 1$, $\binom{2n}{n+1}$, n, $n(n^n - (n-1)^n)$, $\sum_{i<n} \sum_{j<n} \binom{n}{i} \binom{n}{j} (2^i - 1)^j$

the big picture



$2^n - 1,\ \binom{2n}{n+1},\ n,\ n(n^n - (n-1)^n),\ \sum_{i<n}\sum_{j<n}\binom{n}{i}\binom{n}{j}(2^i - 1)^j,$?