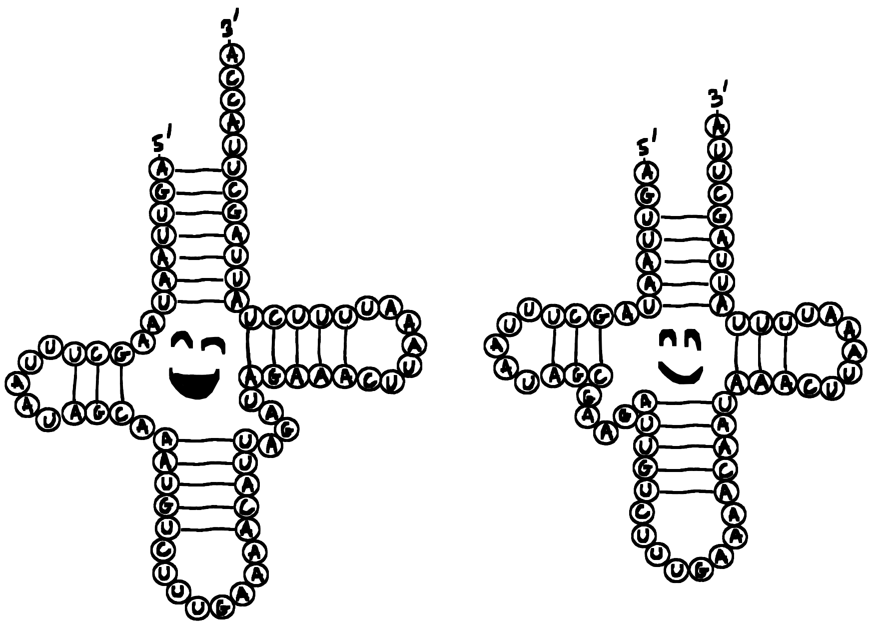
COUNTING, GENERATING AND SAMPLING TREE ALIGNMENTS

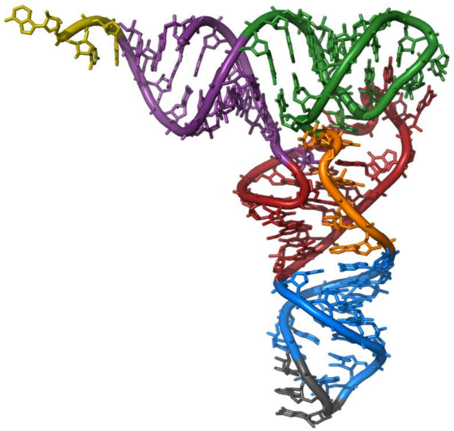
Cedric CHAUVE (Simon Fraser University, Vancouver)
Julien COURTIEL (PIMS/Univ. of British Columbia, Vancouver)
Yann PONTY (CNRS/LIX)

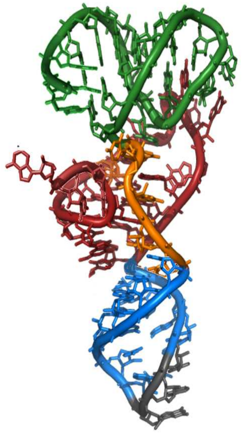


Lille, January 5th 2015

MOTIVATION: RNA COMPARISON

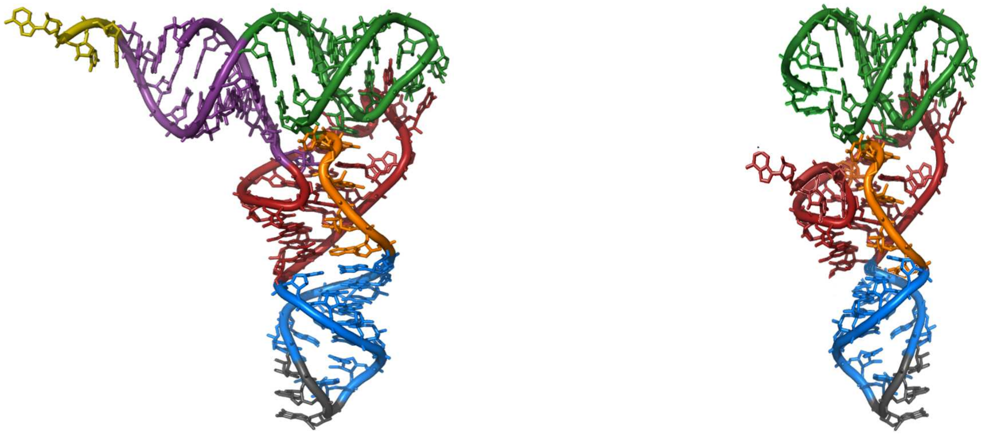
Question: how to measure similarity between two RNAs?





MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?



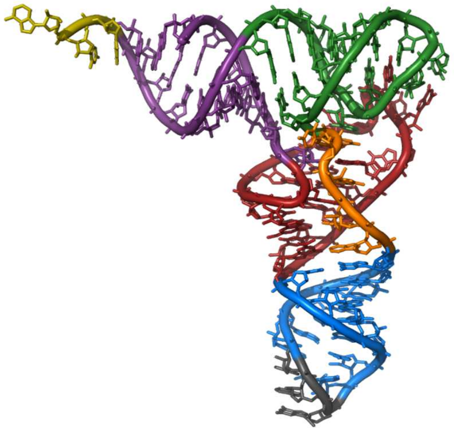
First idea: compare nucleic acid sequences.

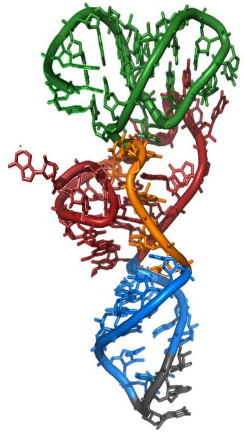
RNA 1: A U U C G A U U A ...

RNA 2: A C C A U G A U U A ...

MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?





First idea: compare nucleic acid sequences.

→ sequence alignment

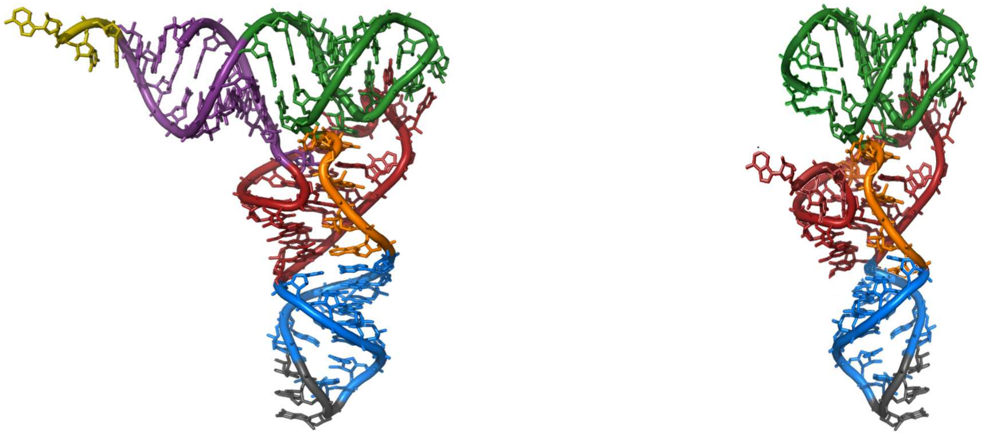
RNA 1: A U U C G A U U A ...

RNA 2: A C C A U G A U U A ...

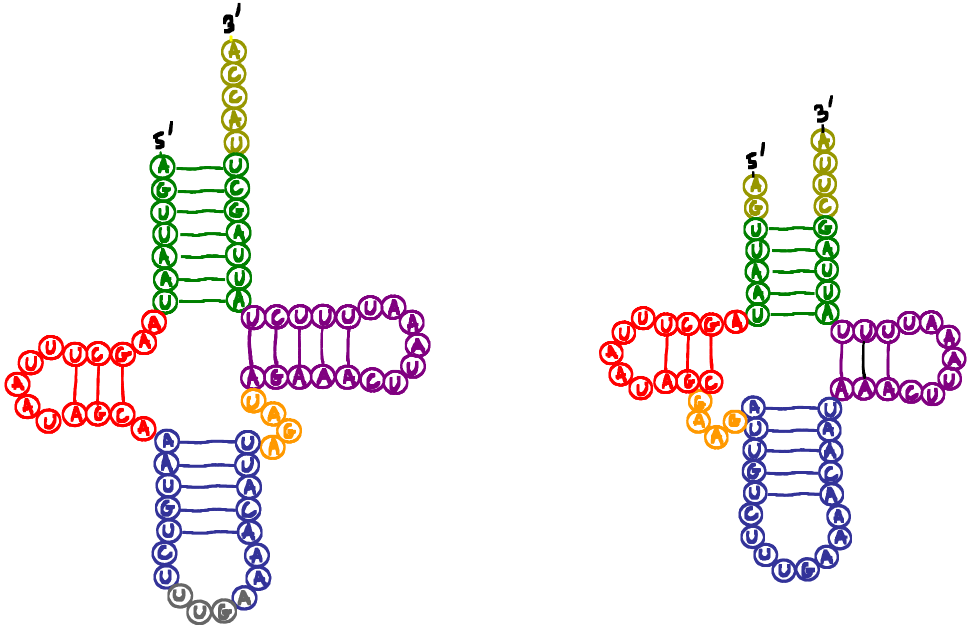
alignment: $\begin{pmatrix}A\\A\end{pmatrix}\begin{pmatrix}U\\-\end{pmatrix}\begin{pmatrix}-\\C\end{pmatrix}\begin{pmatrix}U\\-\end{pmatrix}\begin{pmatrix}C\\C\end{pmatrix}\begin{pmatrix}-\\A\end{pmatrix}\begin{pmatrix}-\\U\end{pmatrix}\begin{pmatrix}G\\G\end{pmatrix}\begin{pmatrix}A\\A\end{pmatrix}\begin{pmatrix}U\\U\end{pmatrix}\begin{pmatrix}U\\U\end{pmatrix}\begin{pmatrix}A\\A\end{pmatrix}\ldots$

MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?

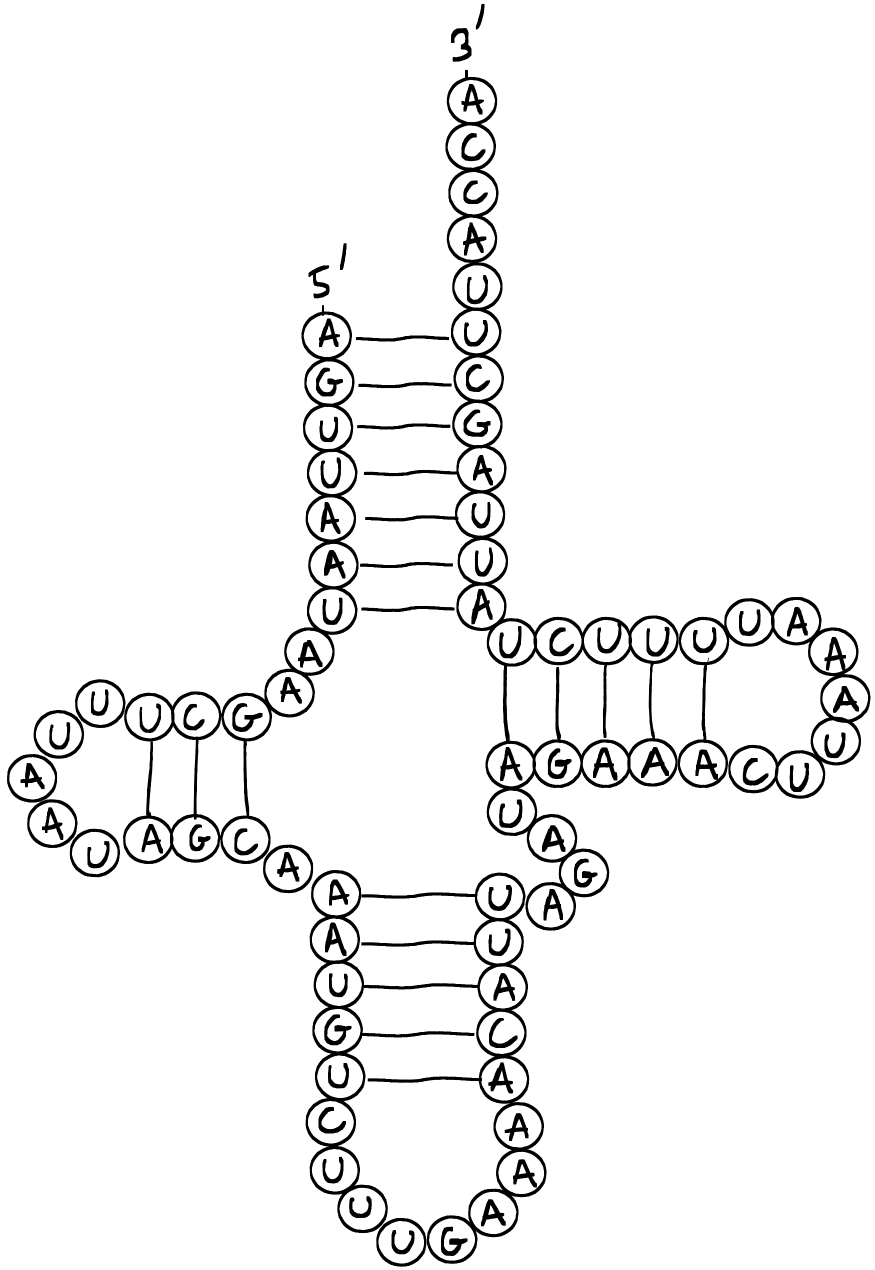


Second idea: compare secondary structures.

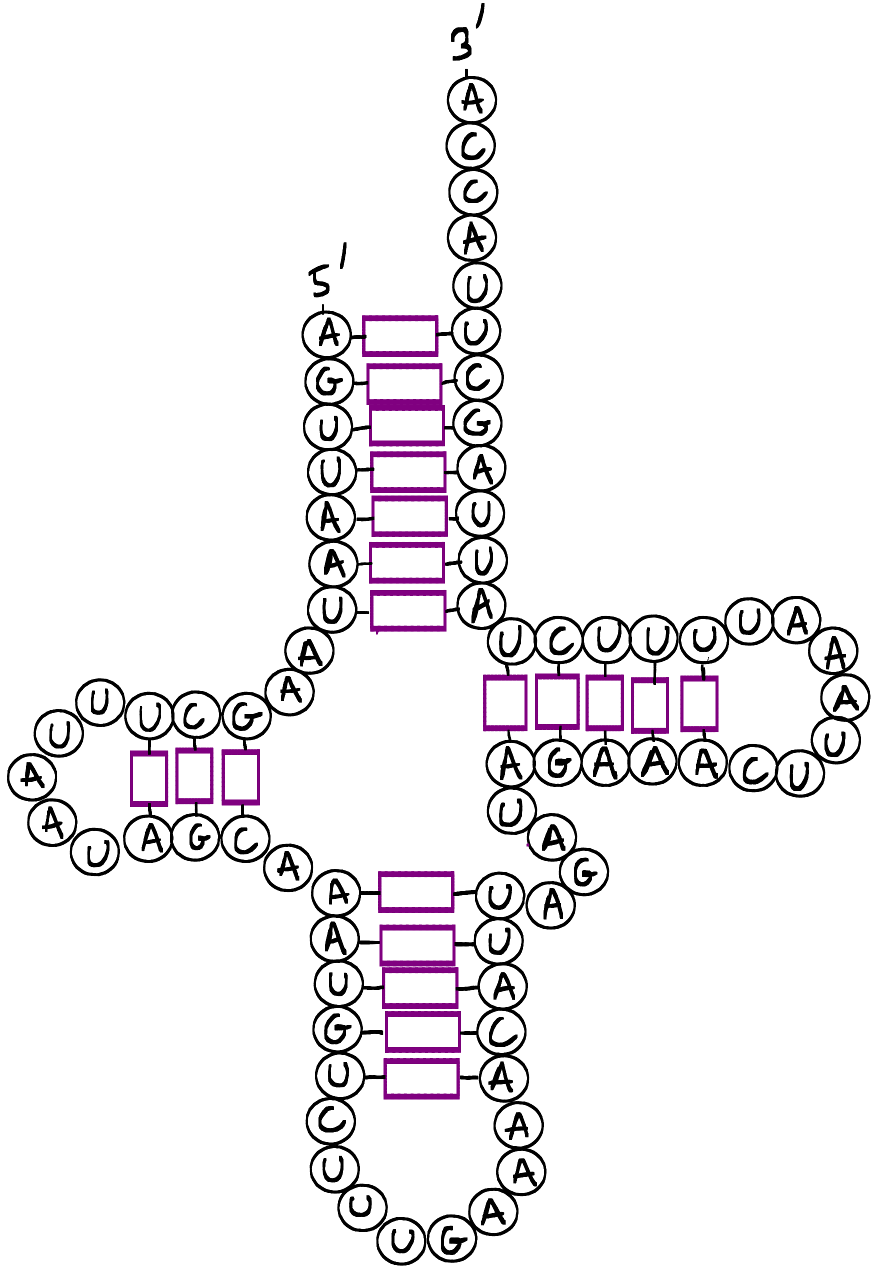


→ notion of tree alignment [Jiang, Wang, Zhang]

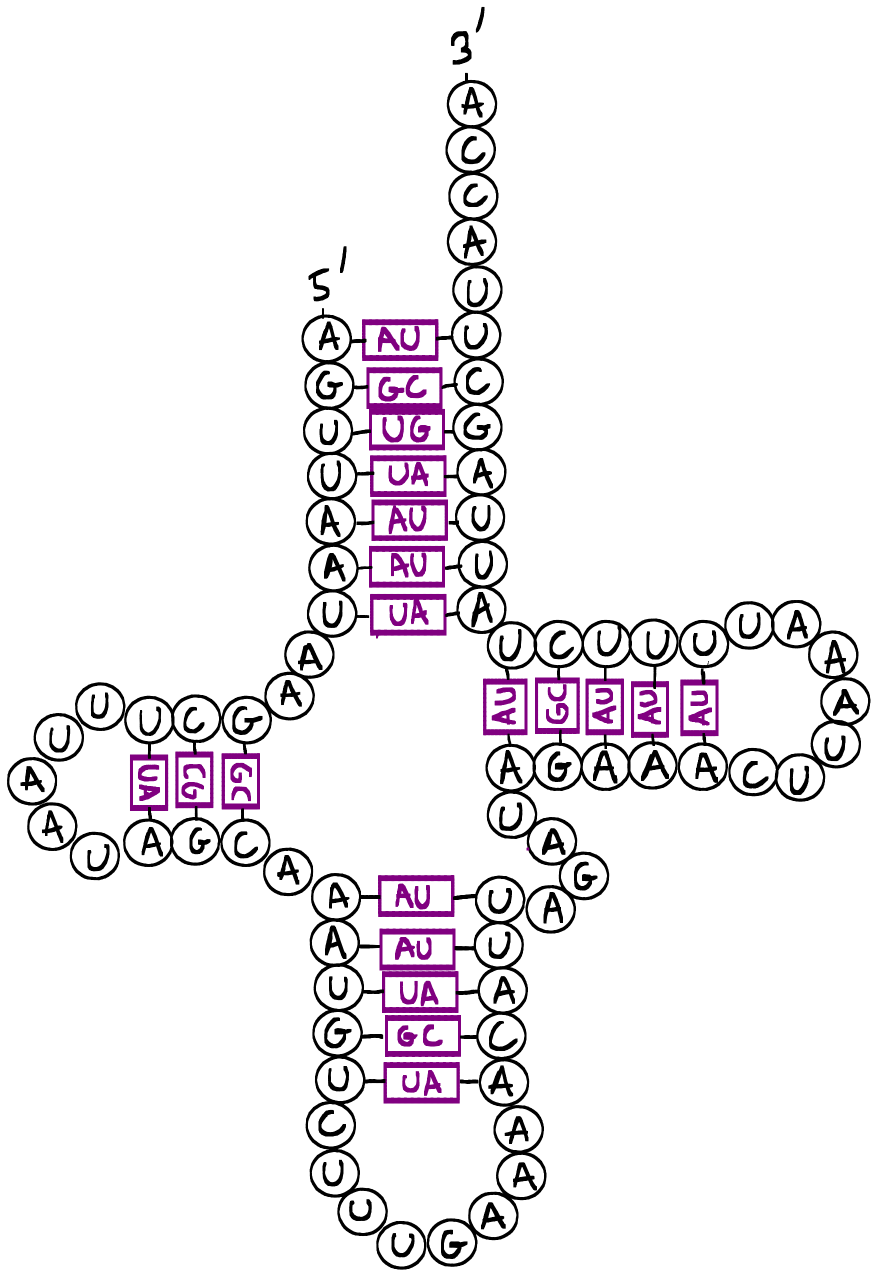
FROM SECONDARY STRUCTURES TO TREES



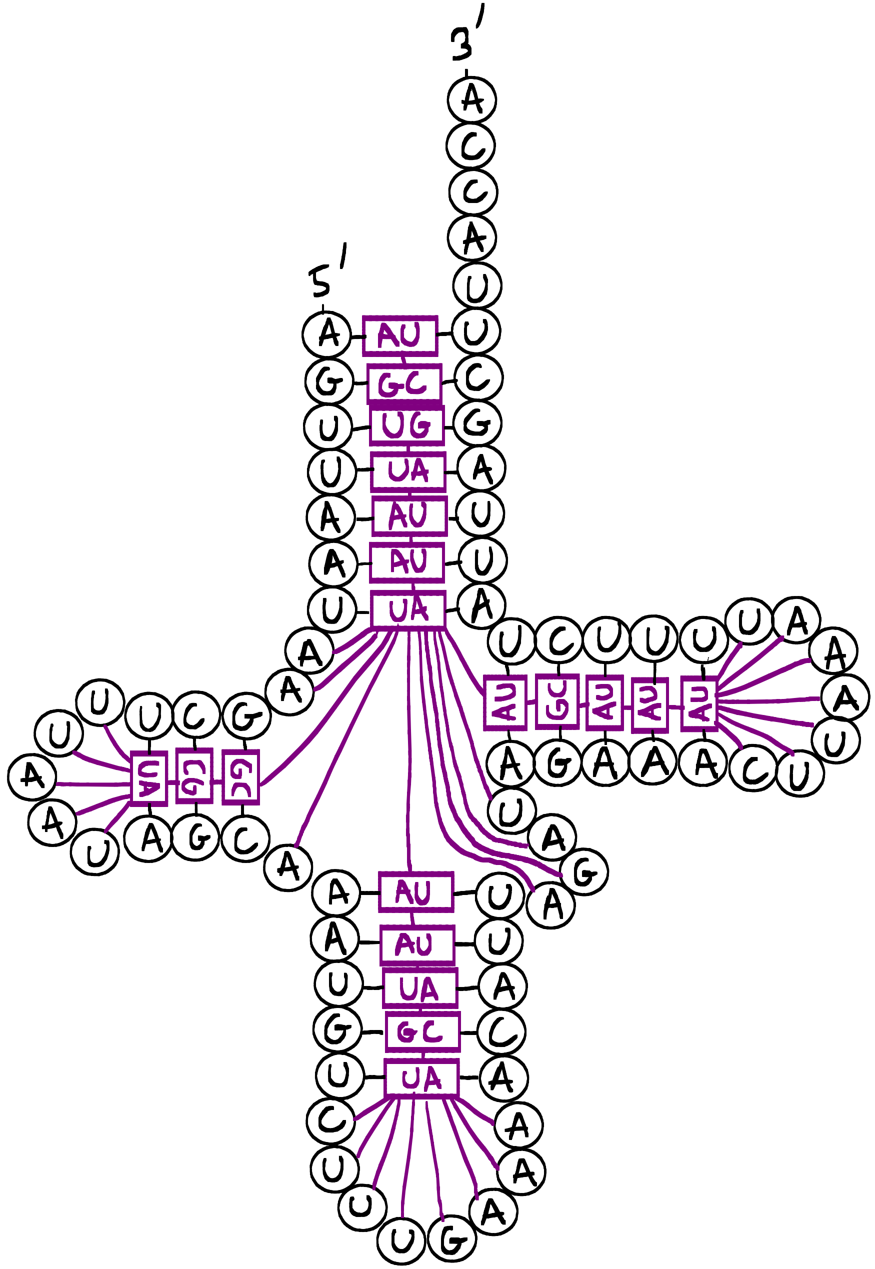
FROM SECONDARY STRUCTURES TO TREES



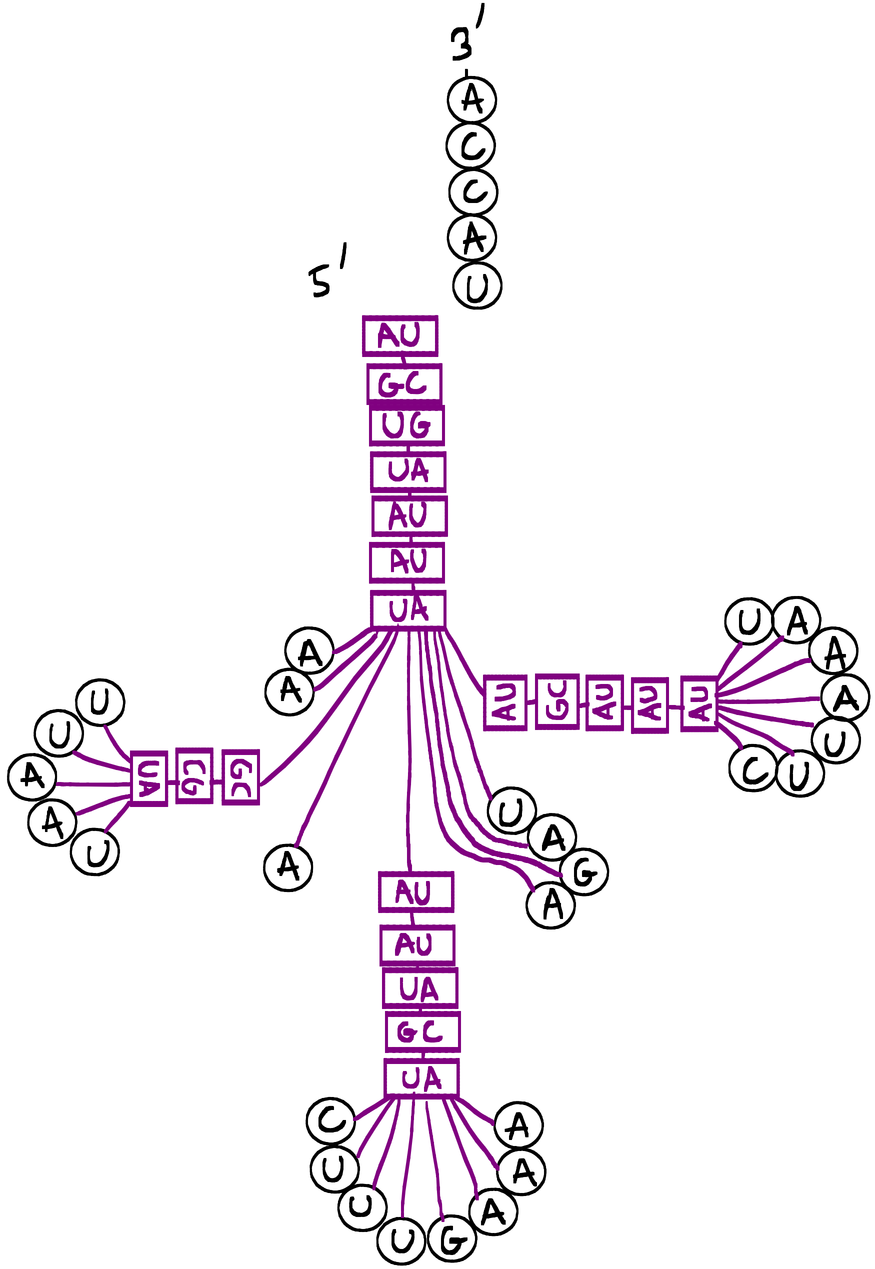
FROM SECONDARY STRUCTURES TO TREES



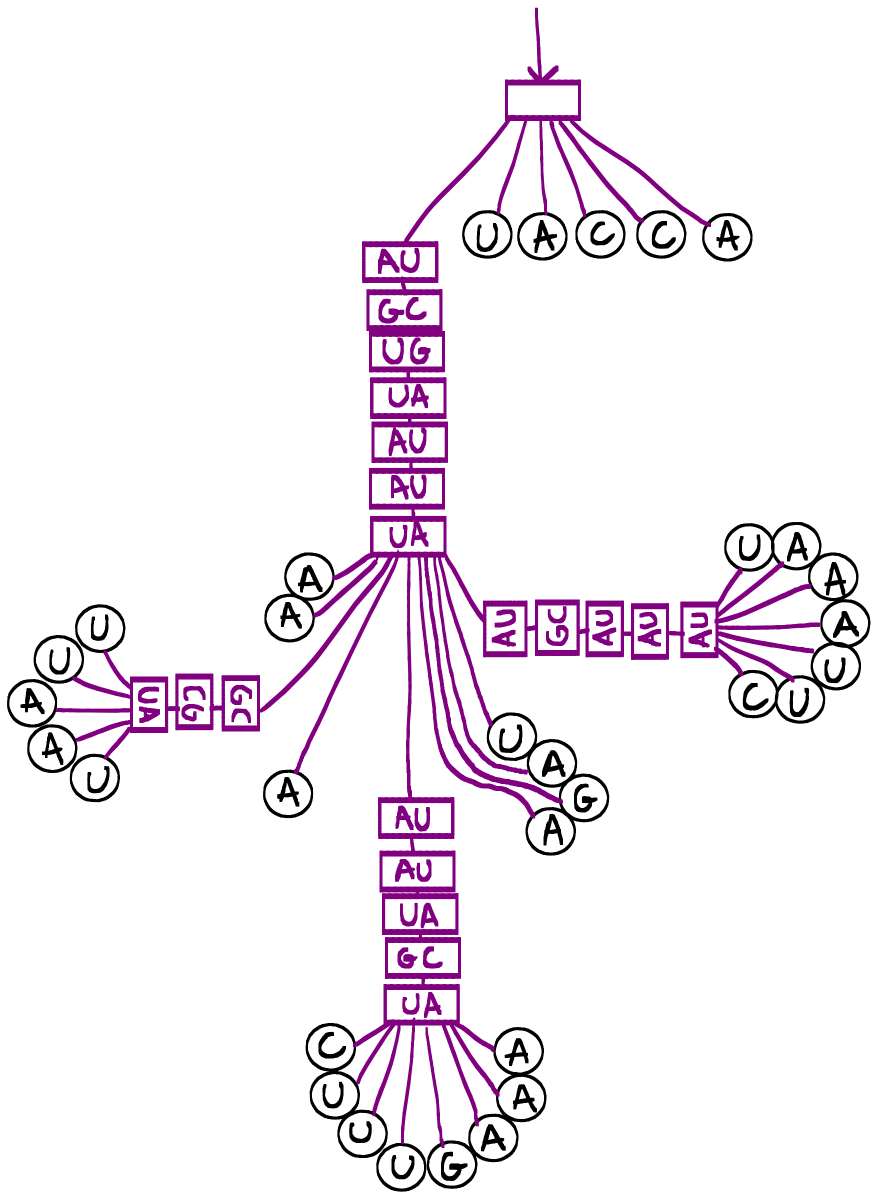
FROM SECONDARY STRUCTURES TO TREES



FROM SECONDARY STRUCTURES TO TREES

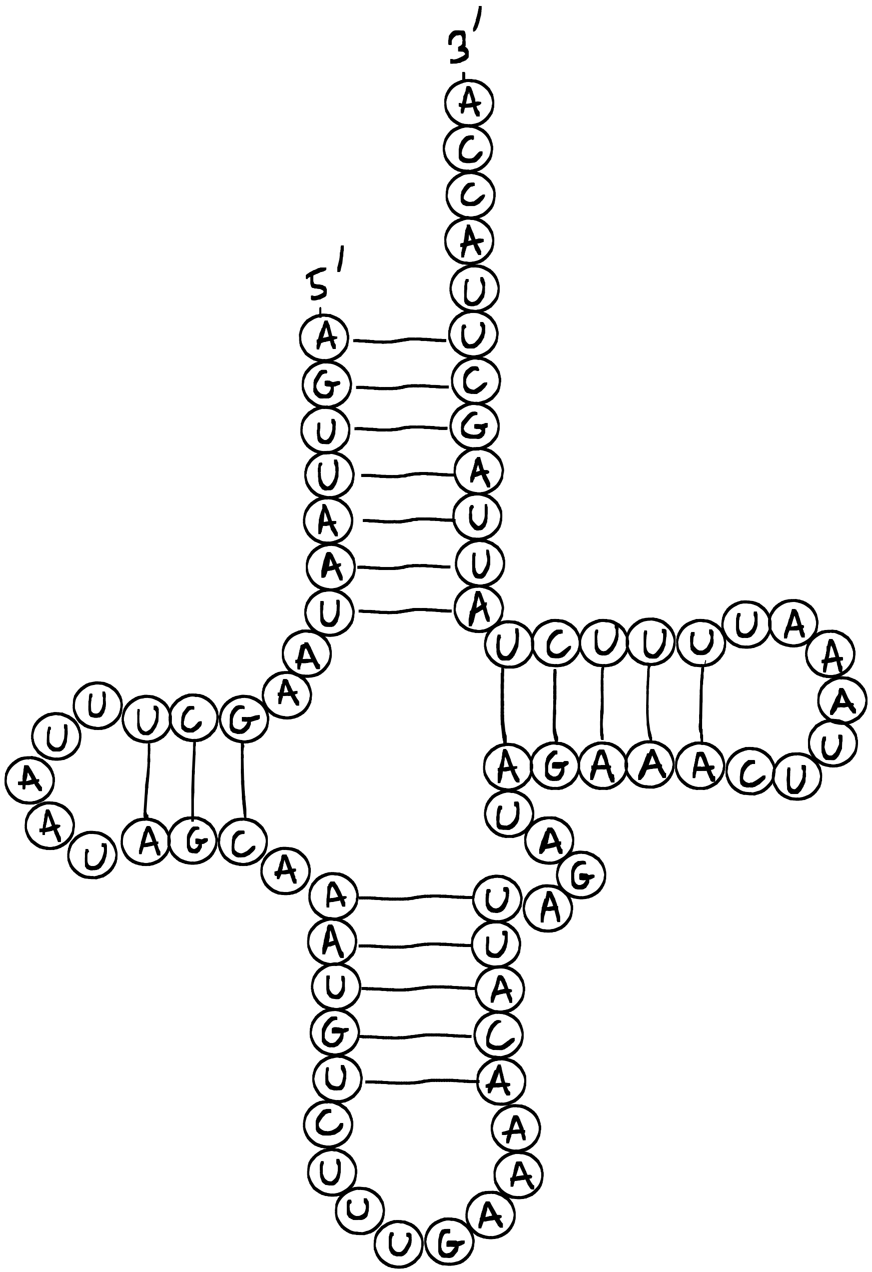


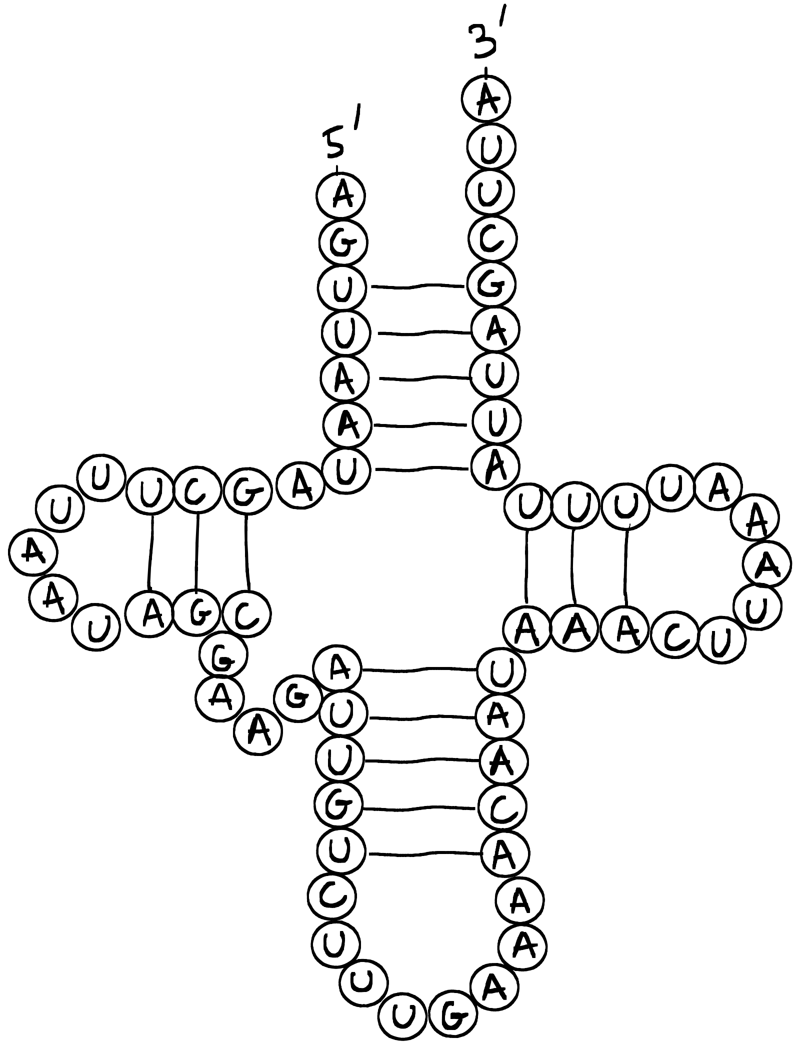
FROM SECONDARY STRUCTURES TO TREES



FROM SECONDARY STRUCTURES TO TREES

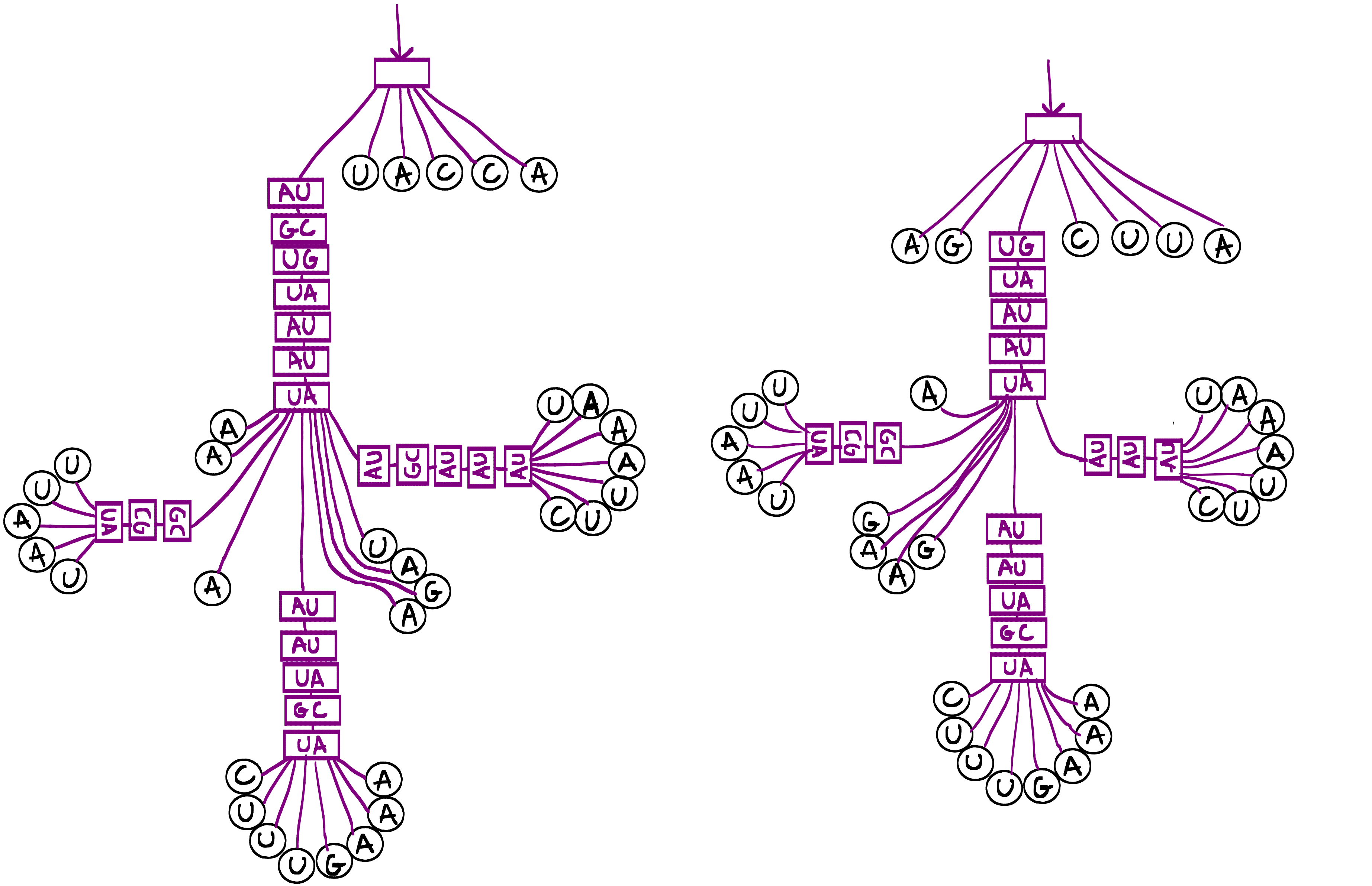
Objective: Align trees coming from RNA 2^{ary} structures





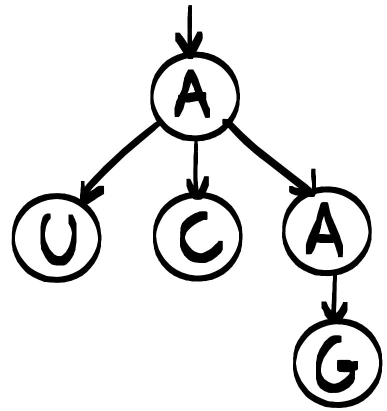
FROM SECONDARY STRUCTURES TO TREES

Objective: Align trees coming from RNA 2^{ary} structures



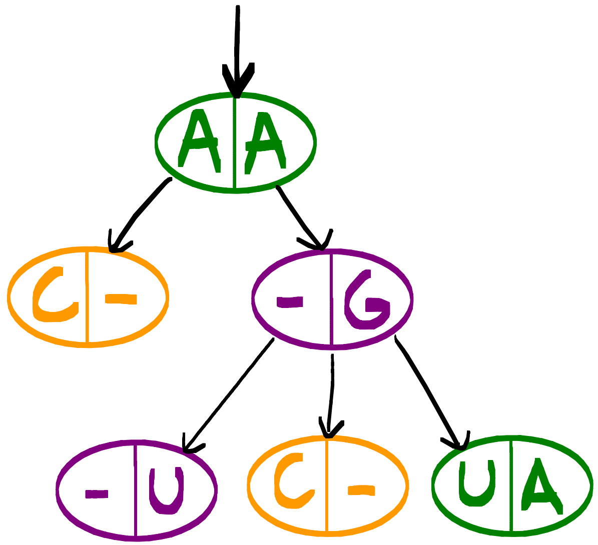
TREES AND SUPERTREES

Trees are plane, rooted,
and vertices are labeled
by an alphabet Σ.



Supertree = tree with
3 types of vertices:

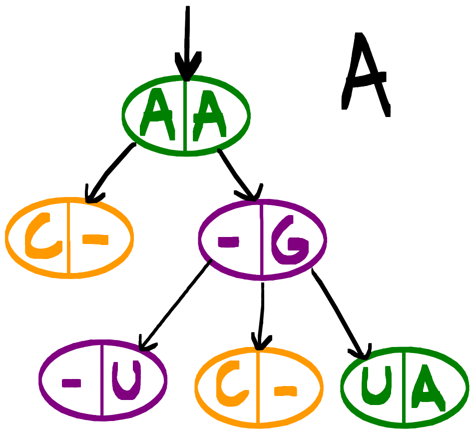
- X | Y (mis)match
- X | - insertion
- - | Y deletion



SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

$\pi_1(A)$ = tree
obtained by changing
X|Y → X
X|- → X
and removing -|Y.



$\pi_2(A)$ = tree
obtained by changing
X|Y → Y
-|Y → Y
and removing X|-.

SUPERTREES INDUCE TREE ALIGNMENTS

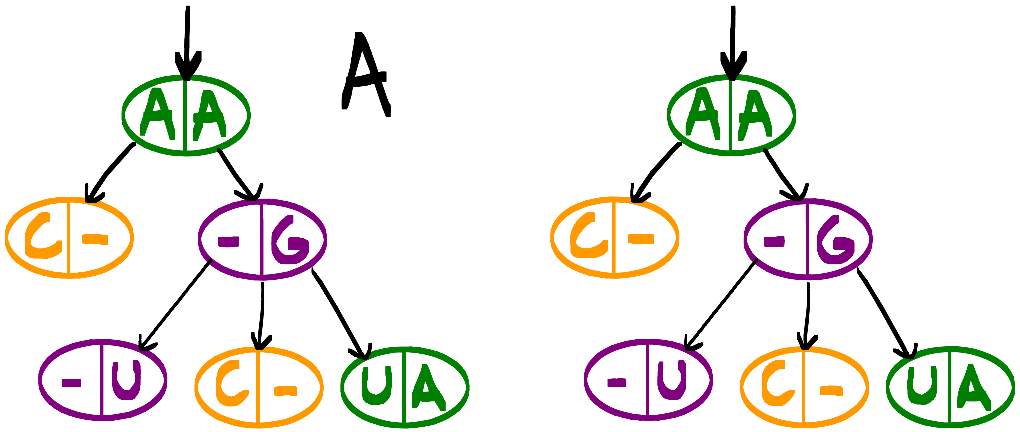
Let A be a supertree,

$\pi_1(A)$ = tree obtained by changing

$X|Y \rightarrow X$

$X|- \rightarrow X$

and removing $-|Y$.



$\pi_2(A)$ = tree obtained by changing

$X|Y \rightarrow Y$

$-|Y \rightarrow Y$

and removing $X|-$.

SUPERTREES INDUCE TREE ALIGNMENTS

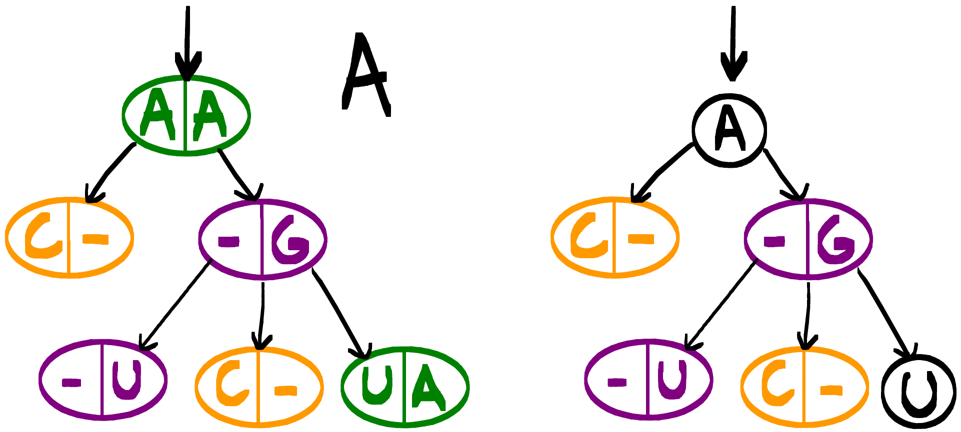
Let A be a supertree,

$\pi_1(A)$ = tree obtained by changing

X|Y → X

X|- → X

and removing -|Y.



$\pi_2(A)$ = tree obtained by changing

X|Y → Y

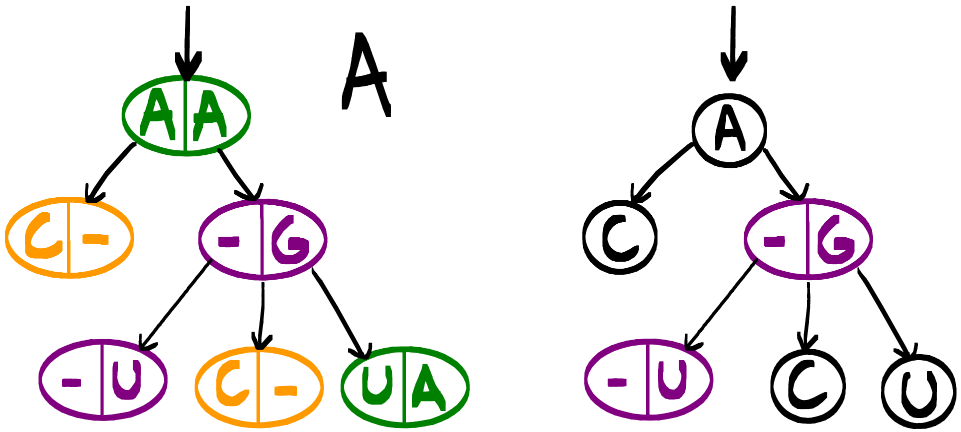
-|Y → Y

and removing X|-.

SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

$\pi_1(A)$ = tree
obtained by changing
X|Y → X
X|- → X
and removing -|Y.



$\pi_2(A)$ = tree
obtained by changing
X|Y → Y
-|Y → Y
and removing X|-.

SUPERTREES INDUCE TREE ALIGNMENTS

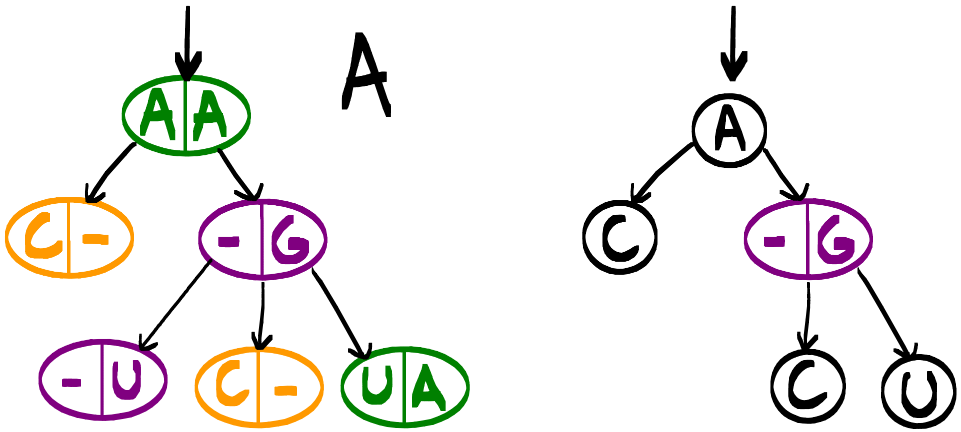
Let A be a supertree,

$\pi_1(A)$ = tree obtained by changing

$X|Y \to X$

$X|- \to X$

and removing $-|Y$.



$\pi_2(A)$ = tree obtained by changing

$X|Y \to Y$

$-|Y \to Y$

and removing $X|-$.

SUPERTREES INDUCE TREE ALIGNMENTS

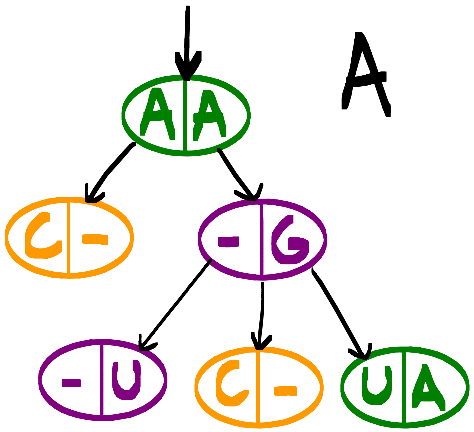
Let A be a supertree,

$\pi_1(A)$ = tree obtained by changing

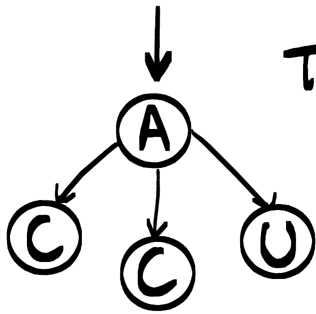
X|Y → X

X|- → X

and removing -|Y.



A



$\pi_1(A)$

$\pi_2(A)$ = tree obtained by changing

X|Y → Y

-|Y → Y

and removing X|-.

SUPERTREES INDUCE TREE ALIGNMENTS

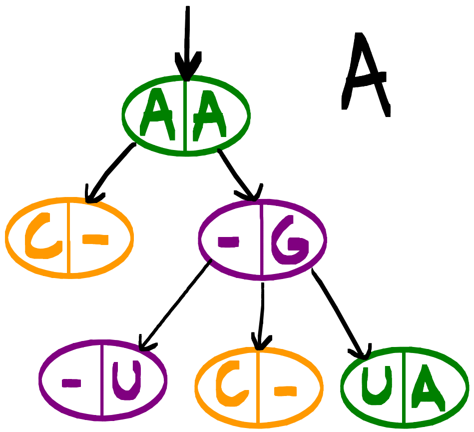
Let A be a supertree,

$\pi_1(A)$ = tree obtained by changing

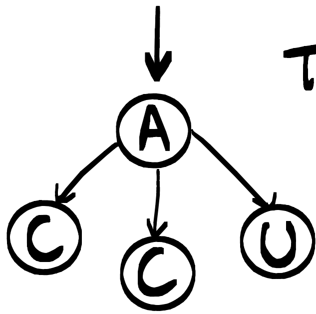
$X|Y \rightarrow X$

$X|- \rightarrow X$

and removing $-|Y$.



A



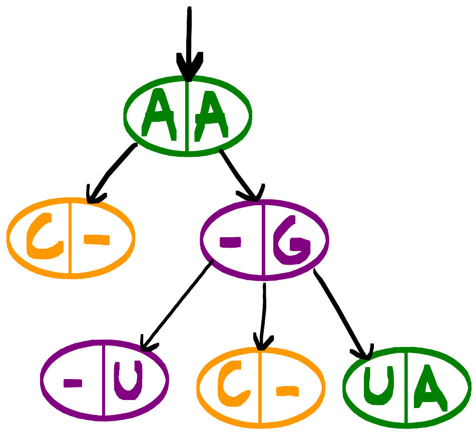
$\pi_1(A)$

$\pi_2(A)$ = tree obtained by changing

$X|Y \rightarrow Y$

$-|Y \rightarrow Y$

and removing $X|-$.



SUPERTREES INDUCE TREE ALIGNMENTS

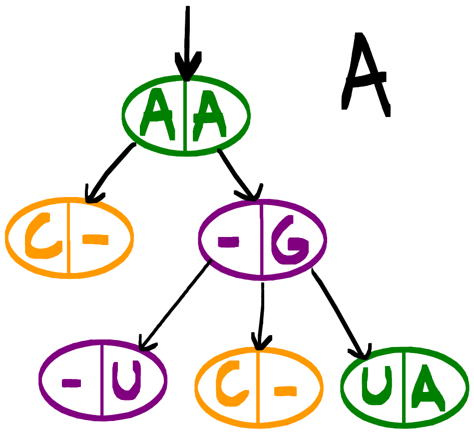
Let A be a supertree,

$\pi_1(A)$ = tree obtained by changing

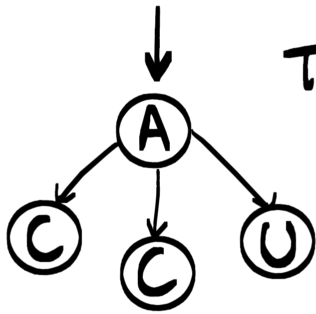
X|Y → X

X|- → X

and removing -|Y.



A



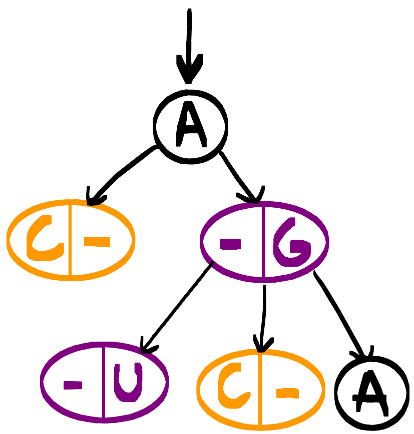
$\pi_1(A)$

$\pi_2(A)$ = tree obtained by changing

X|Y → Y

-|Y → Y

and removing X|-.



SUPERTREES INDUCE TREE ALIGNMENTS

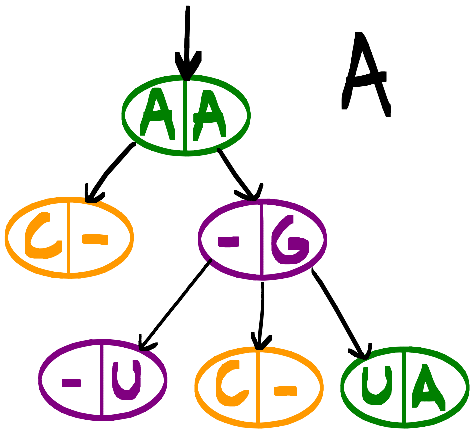
Let A be a supertree,

$\pi_1(A)$ = tree obtained by changing

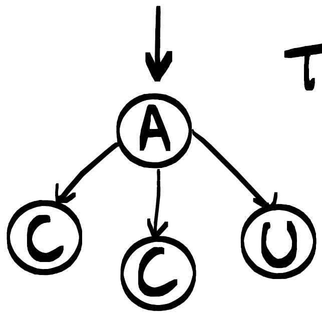
X|Y → X

X|- → X

and removing -|Y.



A



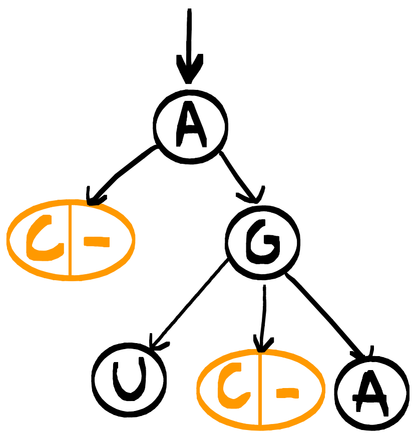
$\pi_1(A)$

$\pi_2(A)$ = tree obtained by changing

X|Y → Y

-|Y → Y

and removing X|-.



SUPERTREES INDUCE TREE ALIGNMENTS

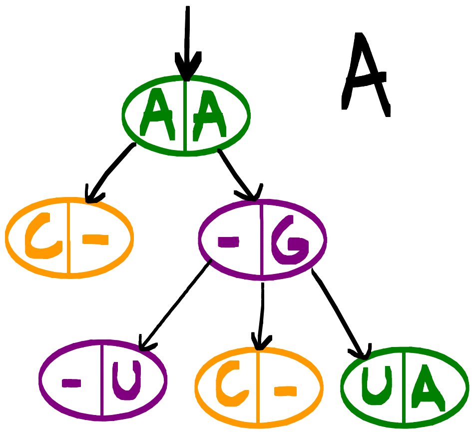
Let A be a supertree,

$\pi_1(A)$ = tree obtained by changing

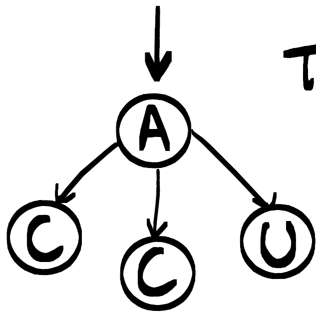
X|Y → X

X|- → X

and removing -|Y.



A



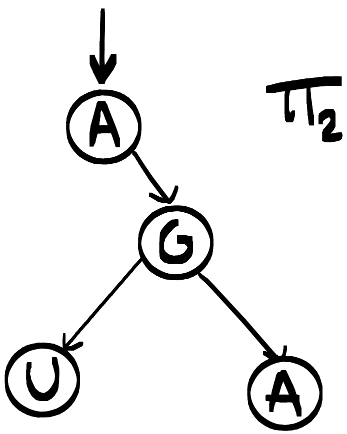
$\pi_1(A)$

$\pi_2(A)$ = tree obtained by changing

X|Y → Y

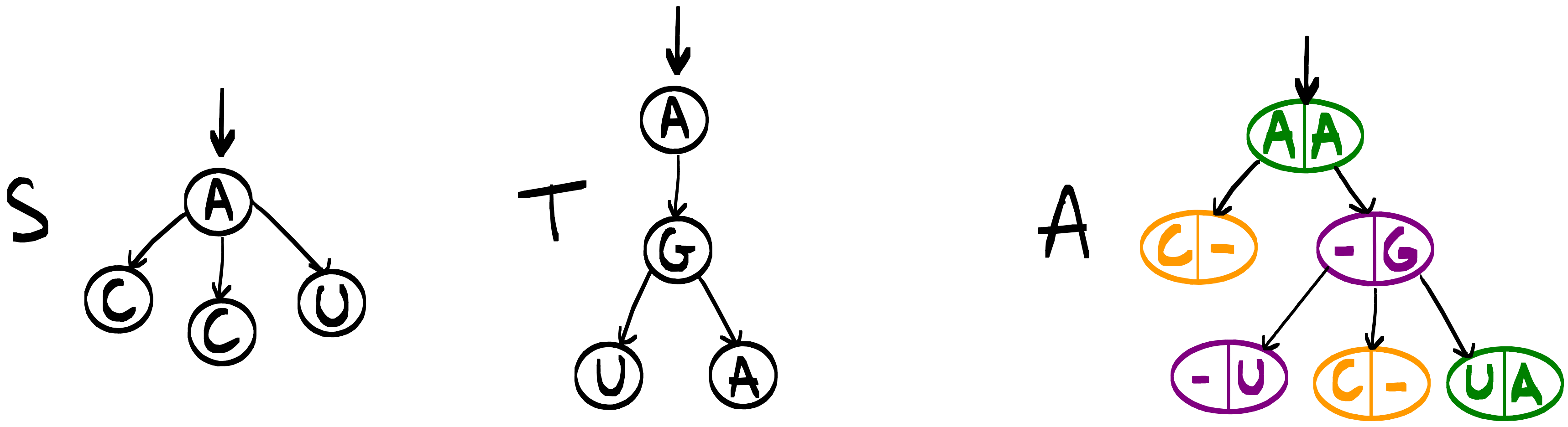
-|Y → Y

and removing X|-.



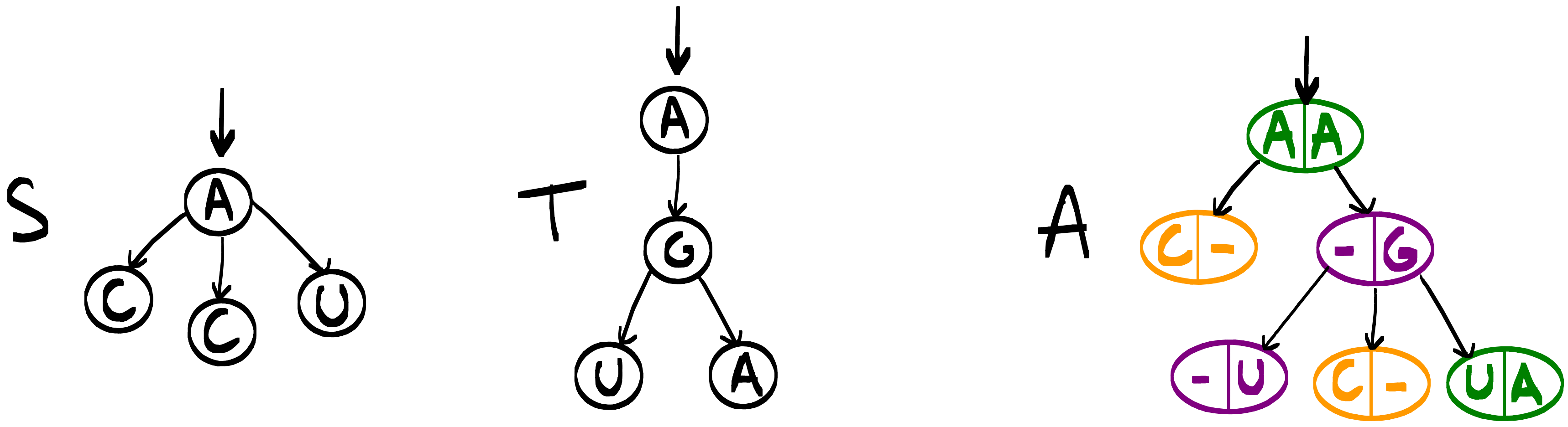
$\pi_2(A)$

SUPERTREES INDUCE TREE ALIGNMENTS



Given two trees S and T,
a supertree A defines an alignment between S and T
if $\pi_1(A) = S$ and $\pi_2(A) = T$.

SUPERTREES INDUCE TREE ALIGNMENTS



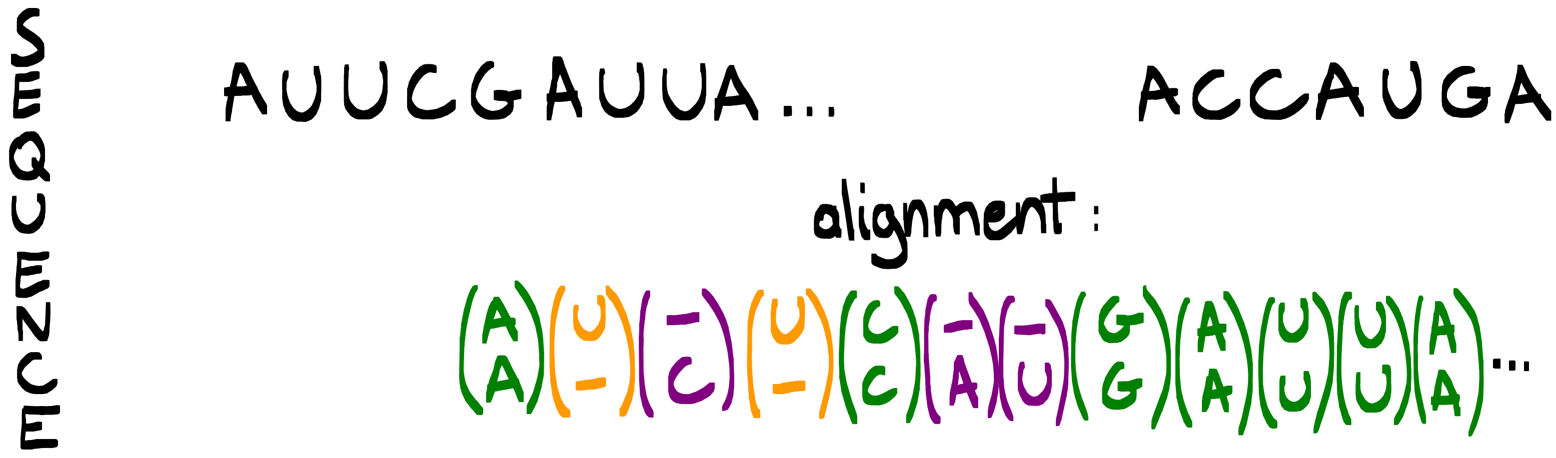
Given two trees S and T,
a supertree A defines an alignment between S and T
if $\pi_1(A) = S$ and $\pi_2(A) = T$.

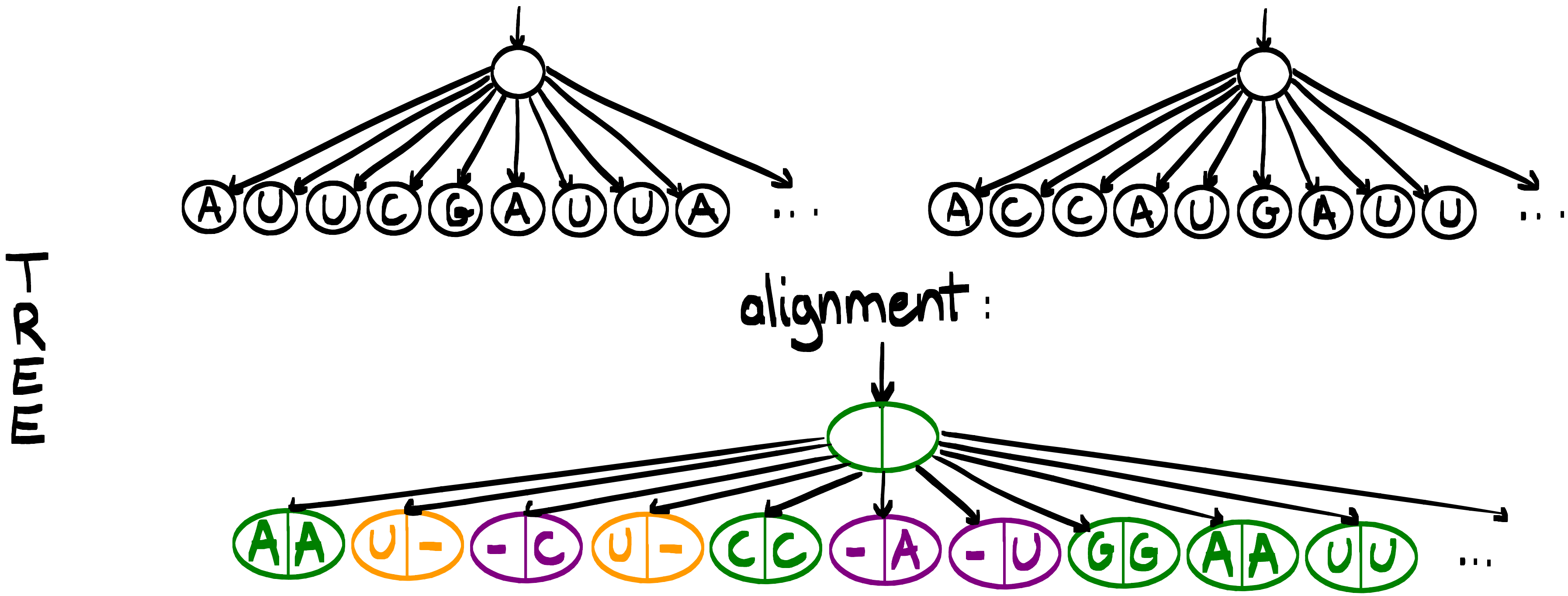
$\text{cost}(A)$ = nb of insertions + deletions + mismatches

(can be changed more complicated models)

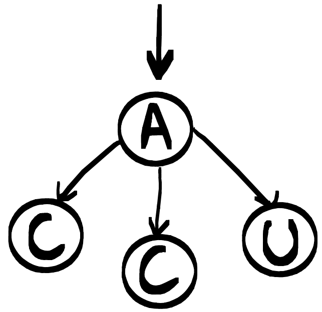
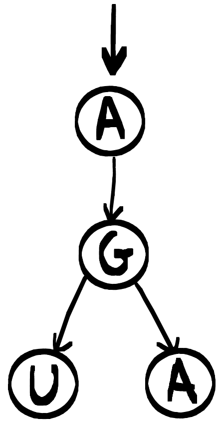
CONNECTION WITH SEQUENCE ALIGNMENTS

Tree alignments generalize sequence alignments.





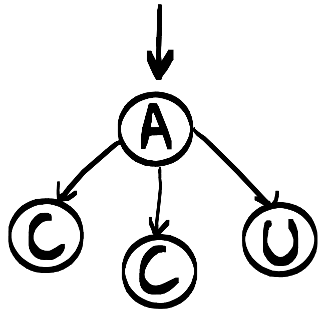
SPACE OF ALIGNMENTS

Which alignment between  and 

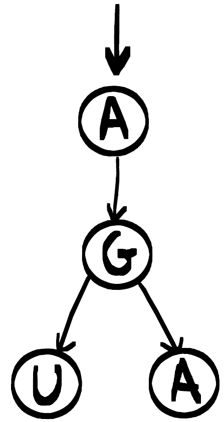
is the most likely?

SPACE OF ALIGNMENTS

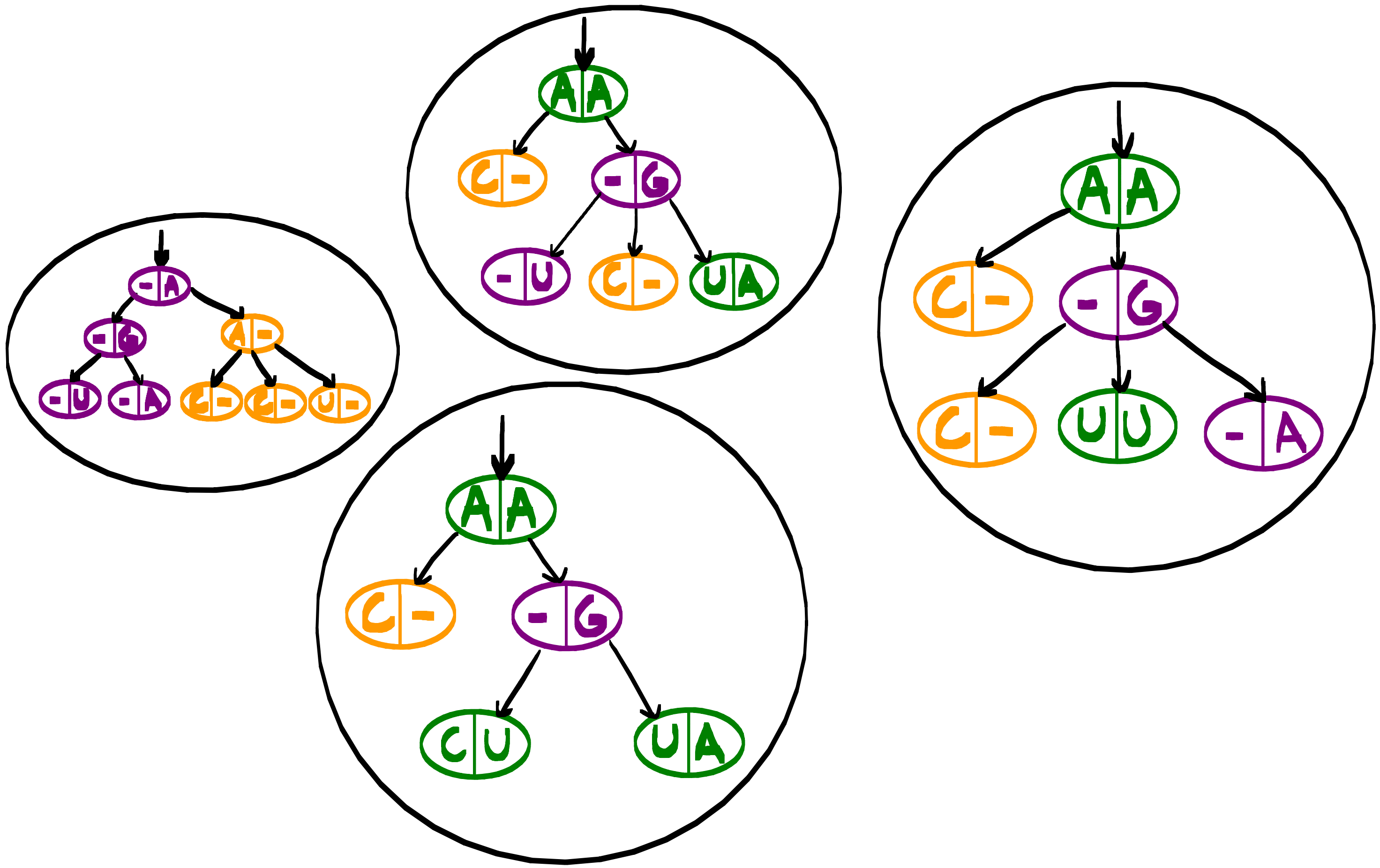
Which alignment between



and



is the most likely?

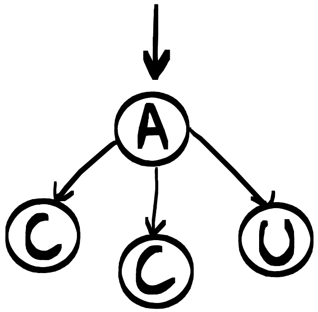
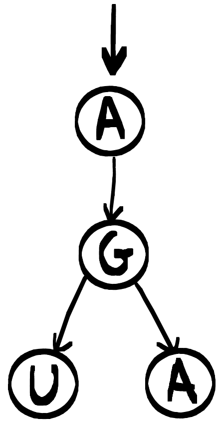


probability of an alignment A

$$\propto e^{-\frac{\mathrm{cost}(A)}{K}}$$

(Gibbs-Boltzmann distribution)

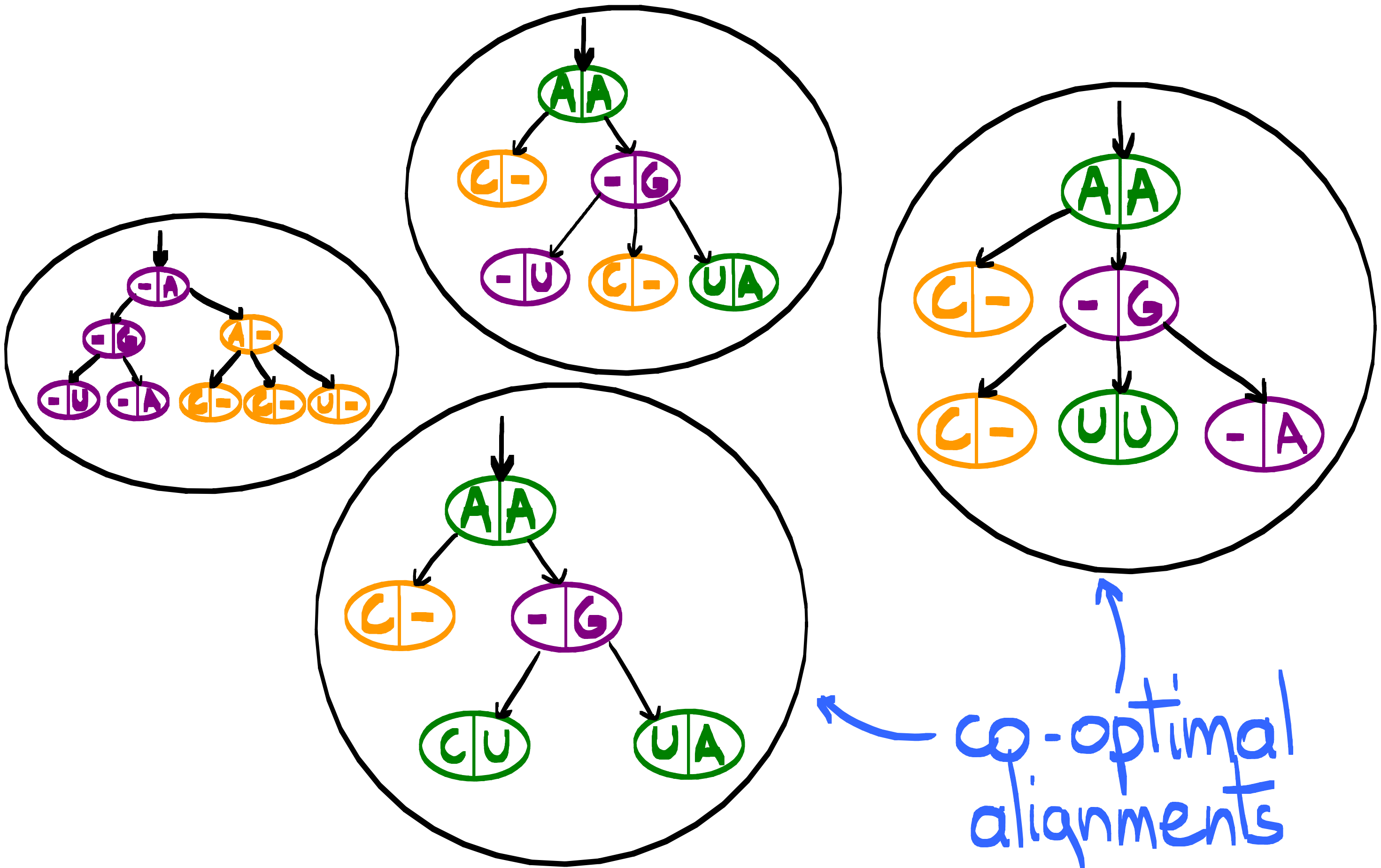
SPACE OF ALIGNMENTS

Which alignment between  and  is the most likely?

probability of an alignment A

$$\propto e^{-\frac{cost(A)}{K}}$$

(Gibbs-Boltzmann distribution)



SPACE OF ALIGNMENTS

Why finding one optimal alignment may be inadequate:

- Co-optimal alignments can be very different.
  (see for instance [Vingron, Argos, 1990])

- Exploring the space of alignments enables the detection of high probability features.

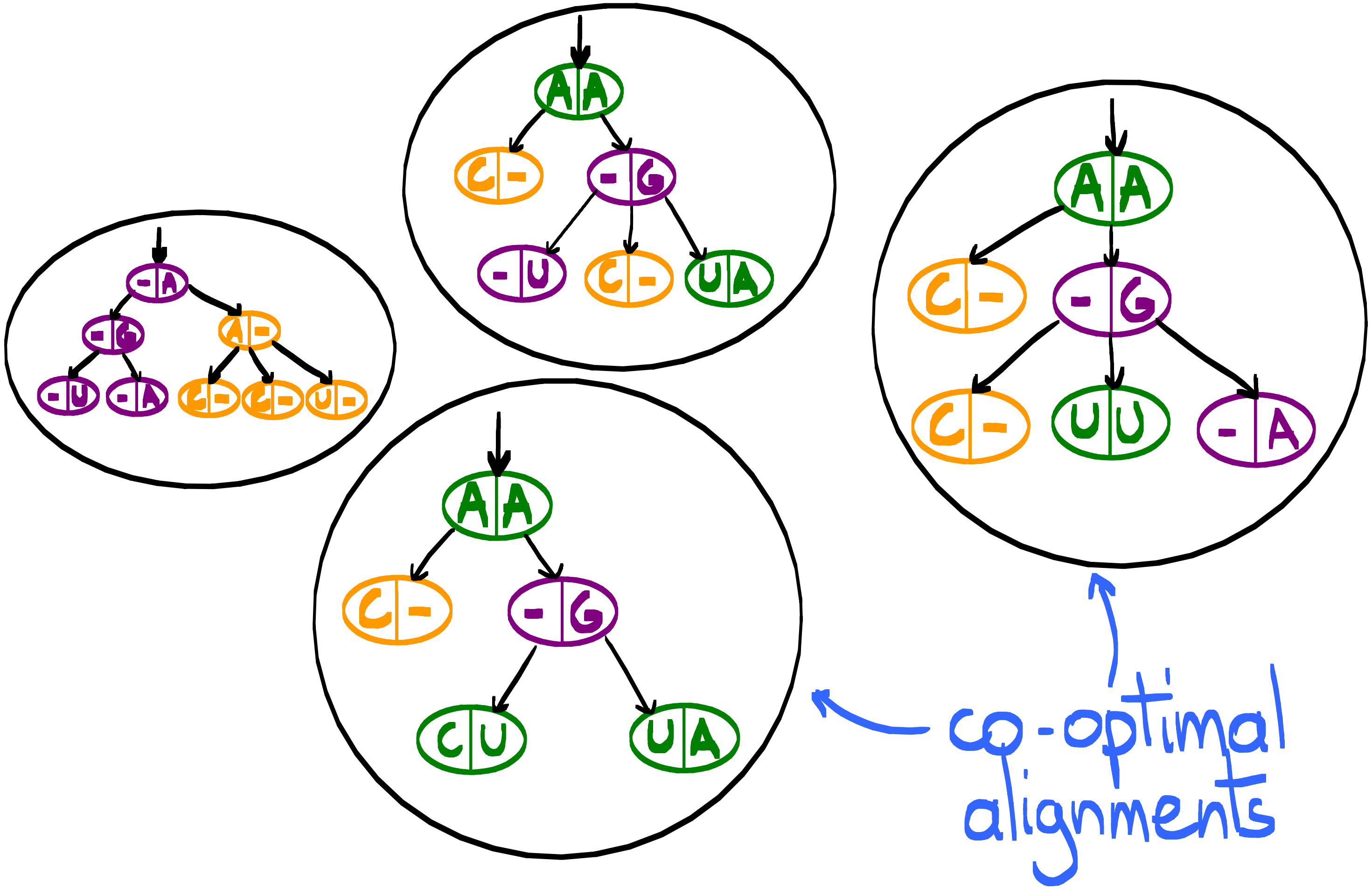
SPACE OF ALIGNMENTS

Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

probability of an alignment A

$$\propto e^{-\frac{cost(A)}{K}}$$

(Gibbs-Boltzmann distribution)

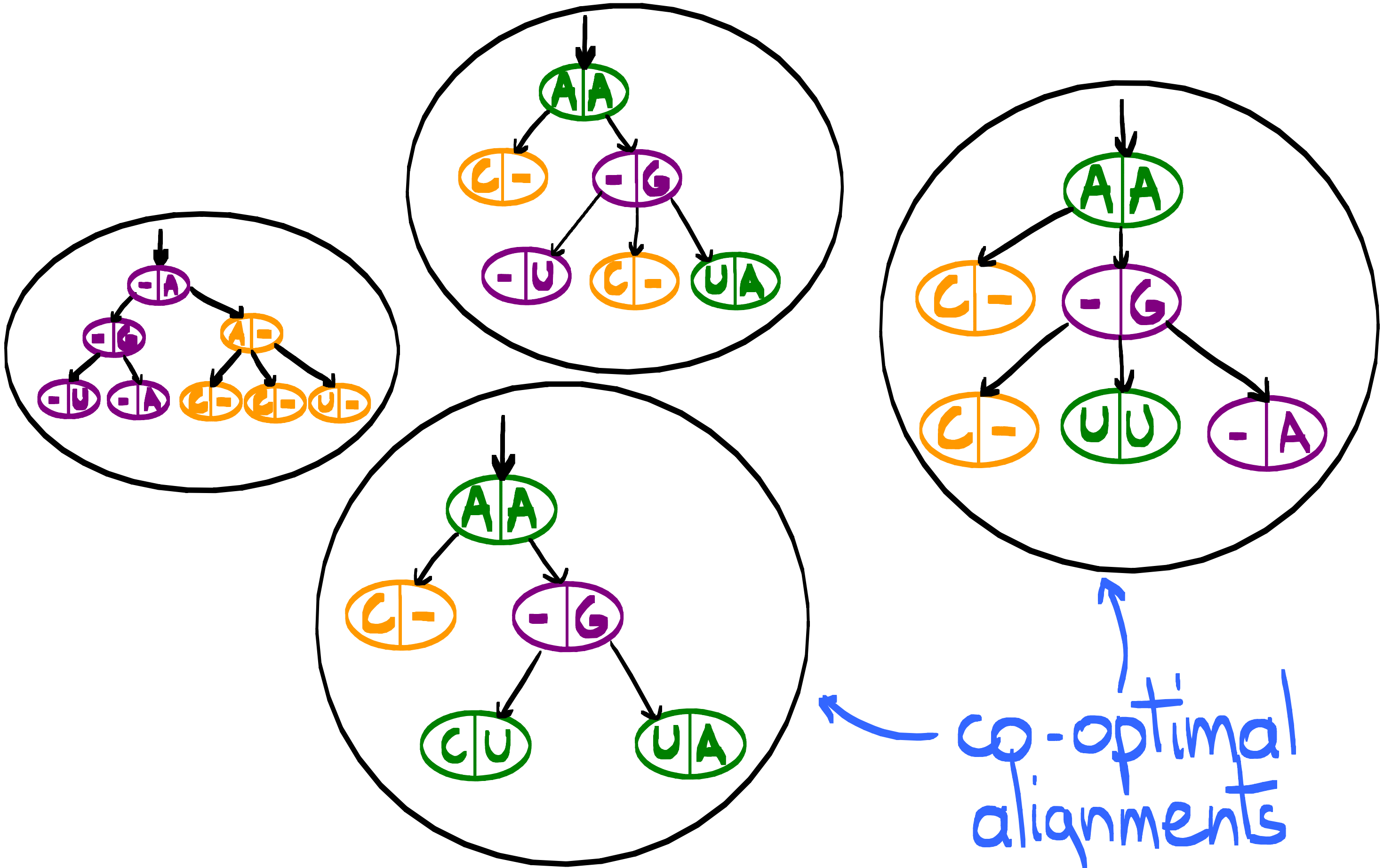


SPACE OF ALIGNMENTS

Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

?

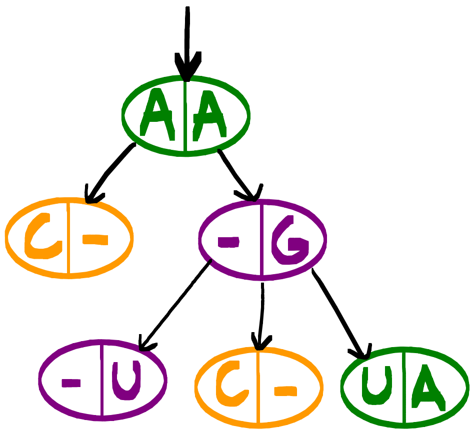
probability of an alignment A $\propto e^{-\frac{cost(A)}{K}}$ (Gibbs-Boltzmann distribution)

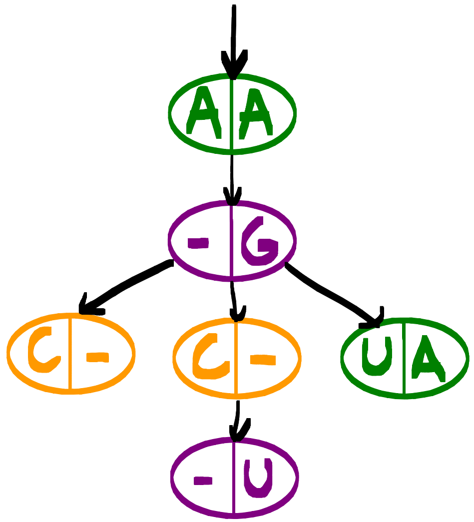


co-optimal alignments

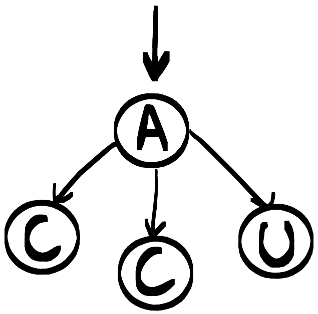
AMBIGUITY OF ALIGNMENTS

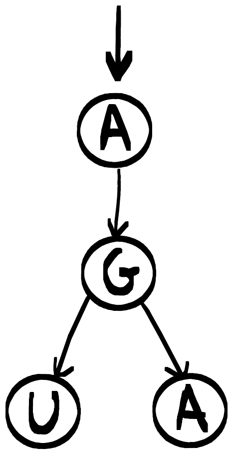
The two supertrees





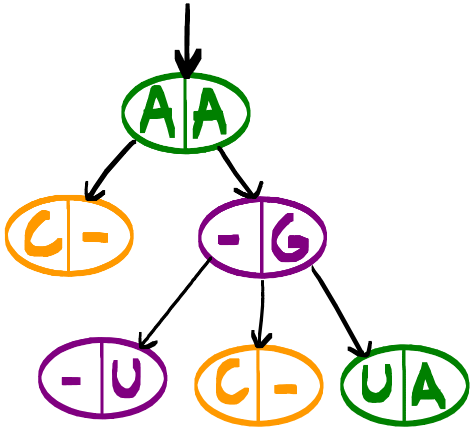
induce the same alignment between the trees

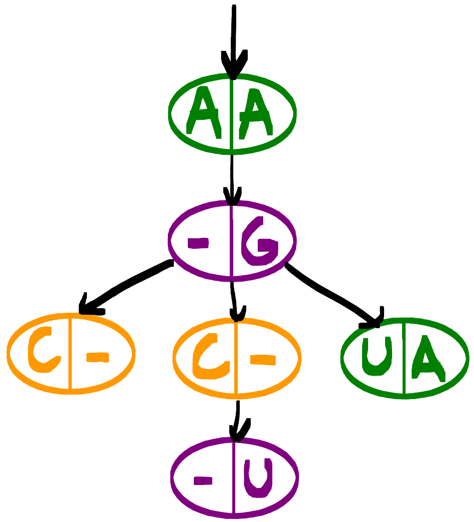




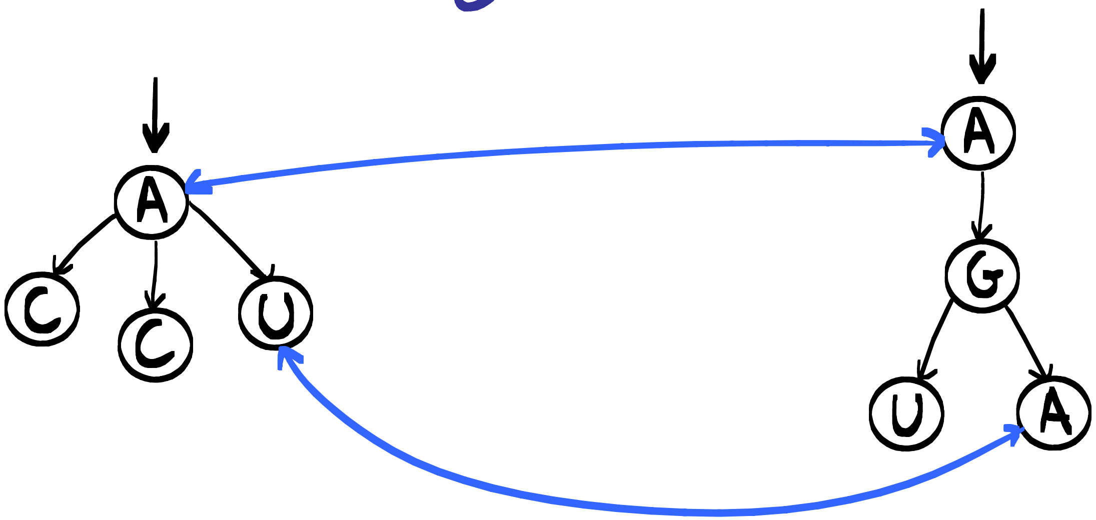
AMBIGUITY OF ALIGNMENTS

The two supertrees



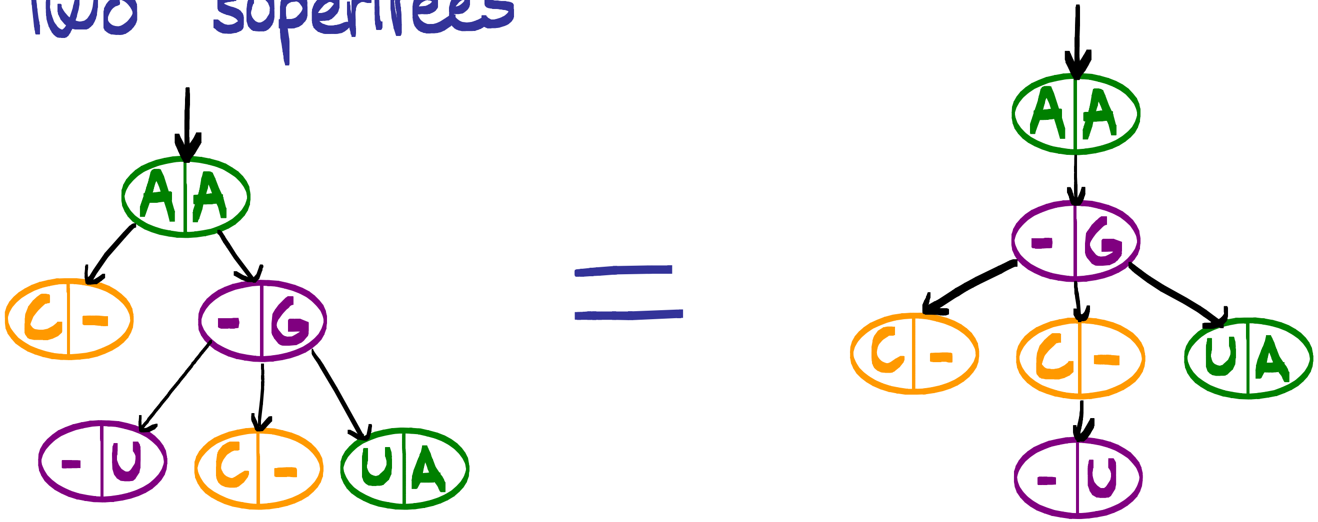


induce the same alignment between the trees

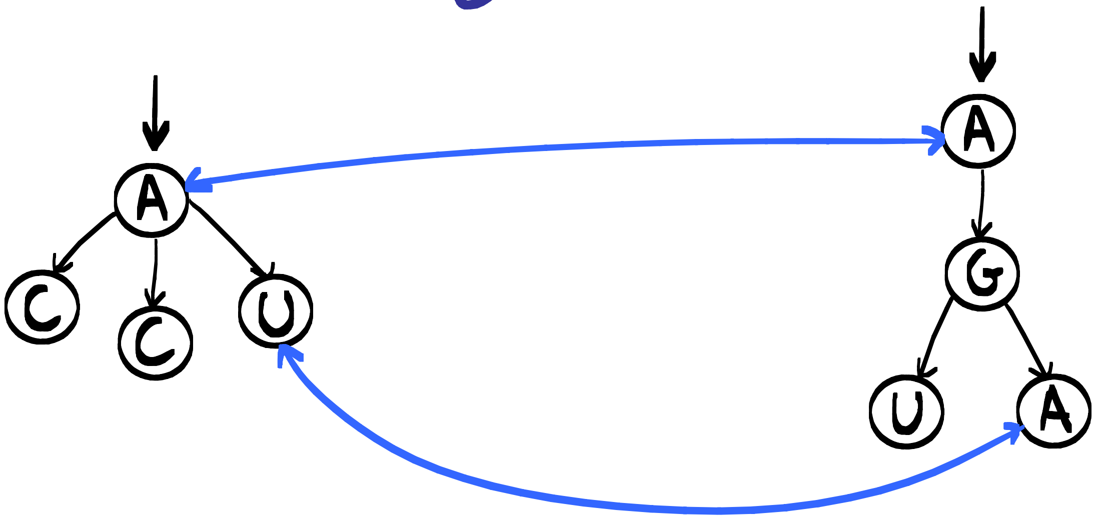


AMBIGUITY OF ALIGNMENTS

The two supertrees



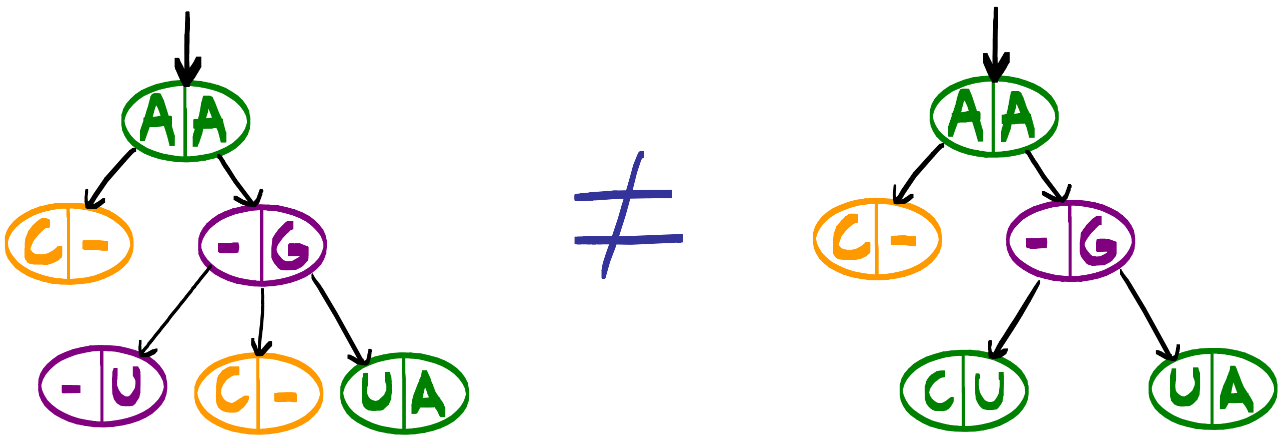
induce the same alignment between the trees



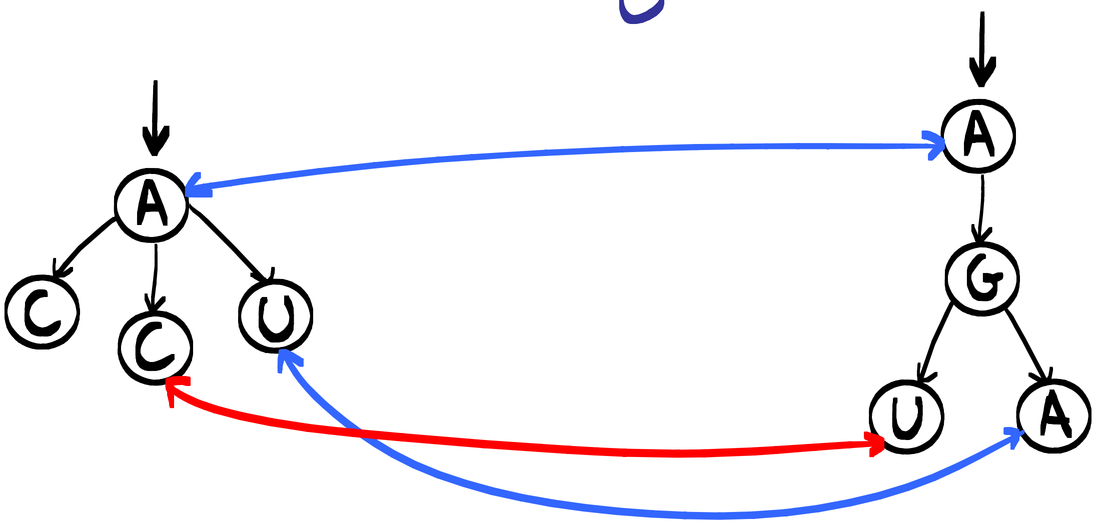
They are the same!

AMBIGUITY OF ALIGNMENTS

The two supertrees



do not induce the same alignment between the trees



A GRAMMAR FOR ALIGNMENTS

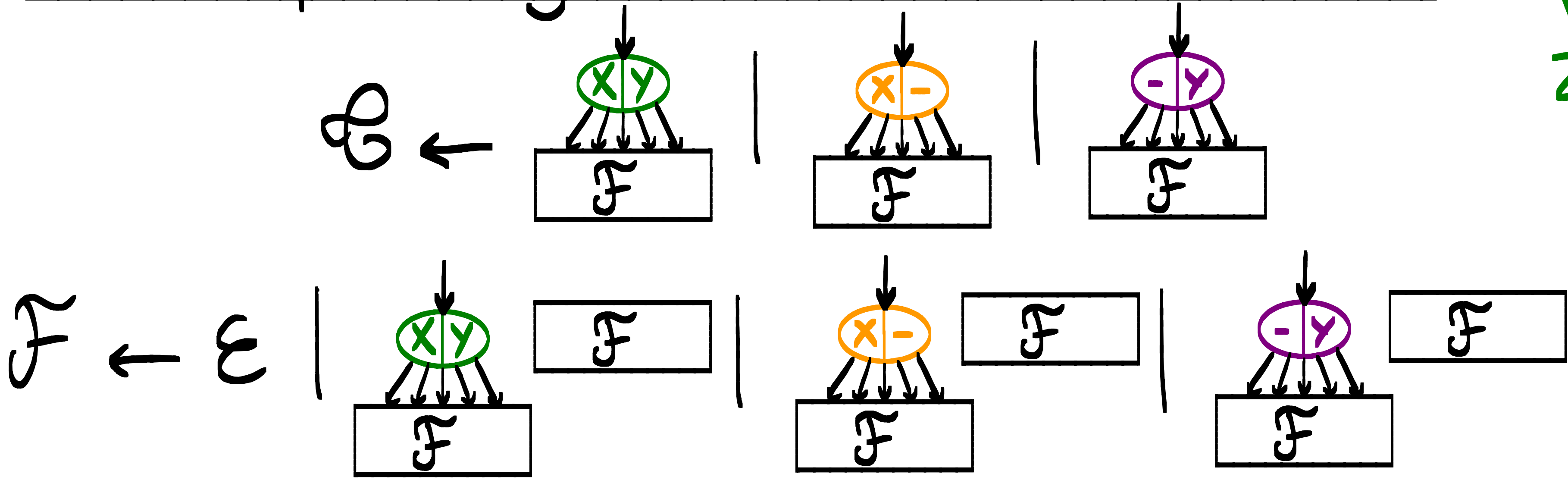
Strategy: Build a context-free grammar that generates every alignment exactly once

A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]

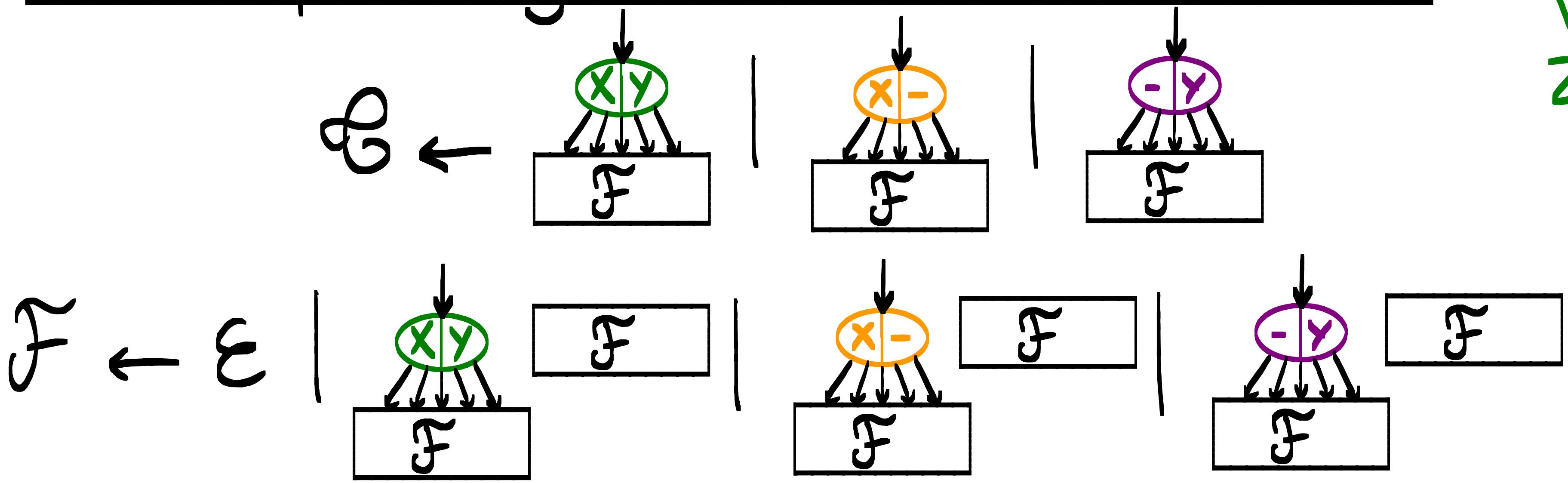


A GRAMMAR FOR ALIGNMENTS

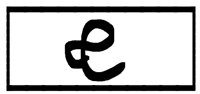
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:

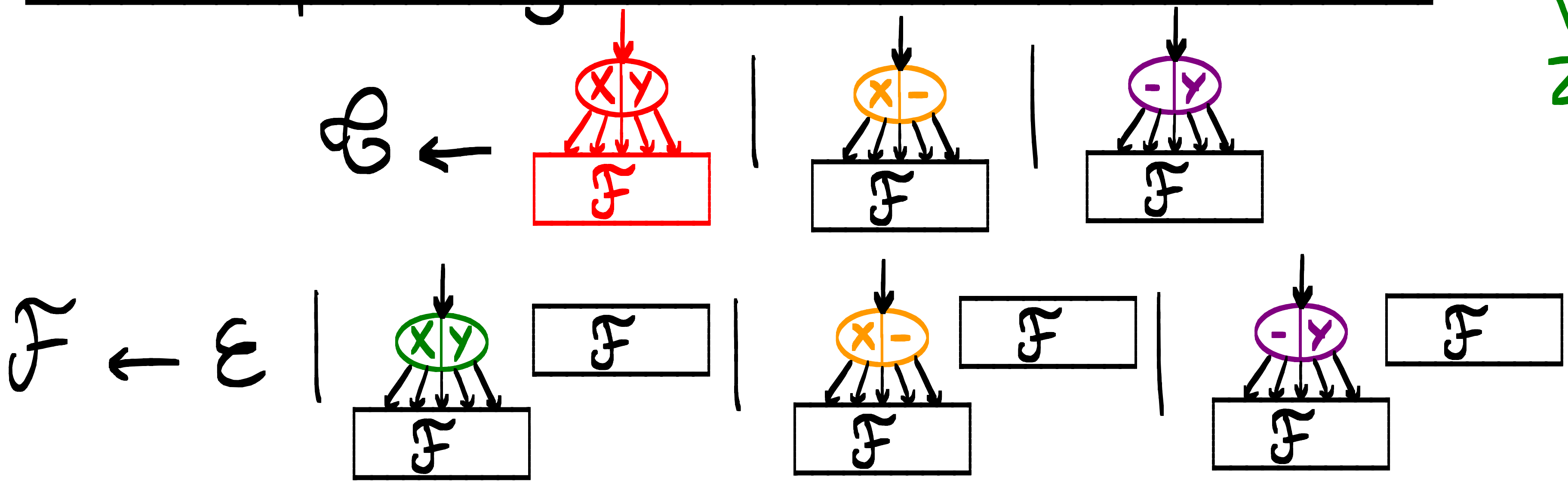


A GRAMMAR FOR ALIGNMENTS

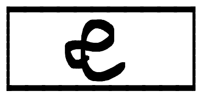
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:

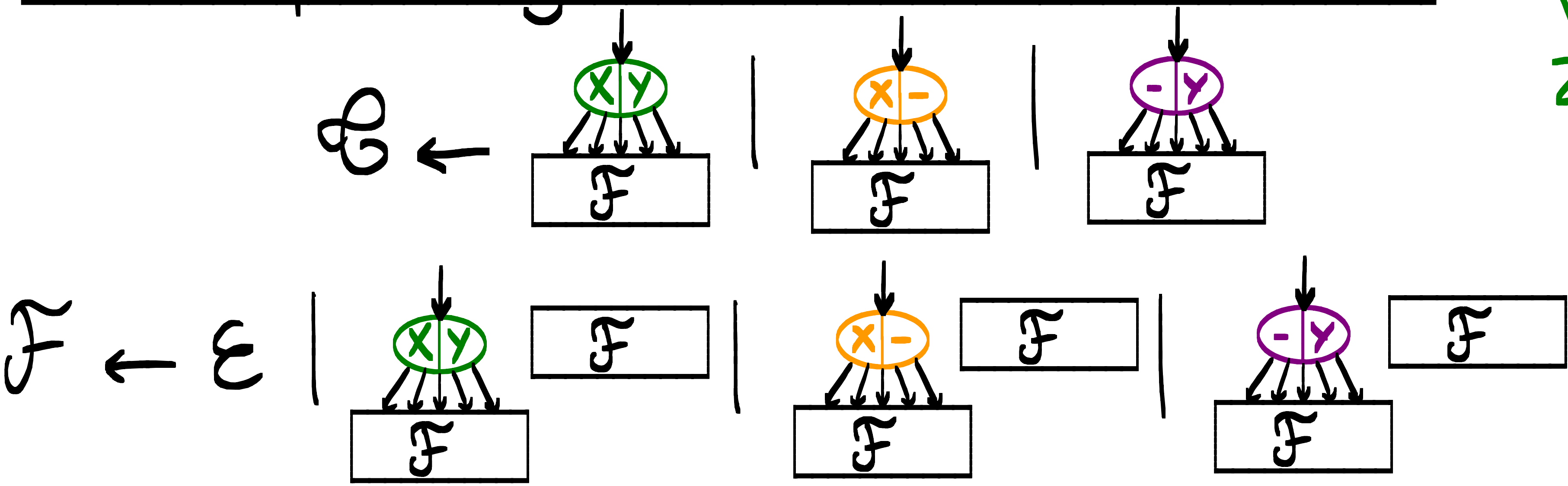


A GRAMMAR FOR ALIGNMENTS

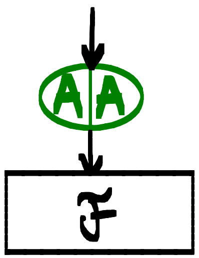
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:

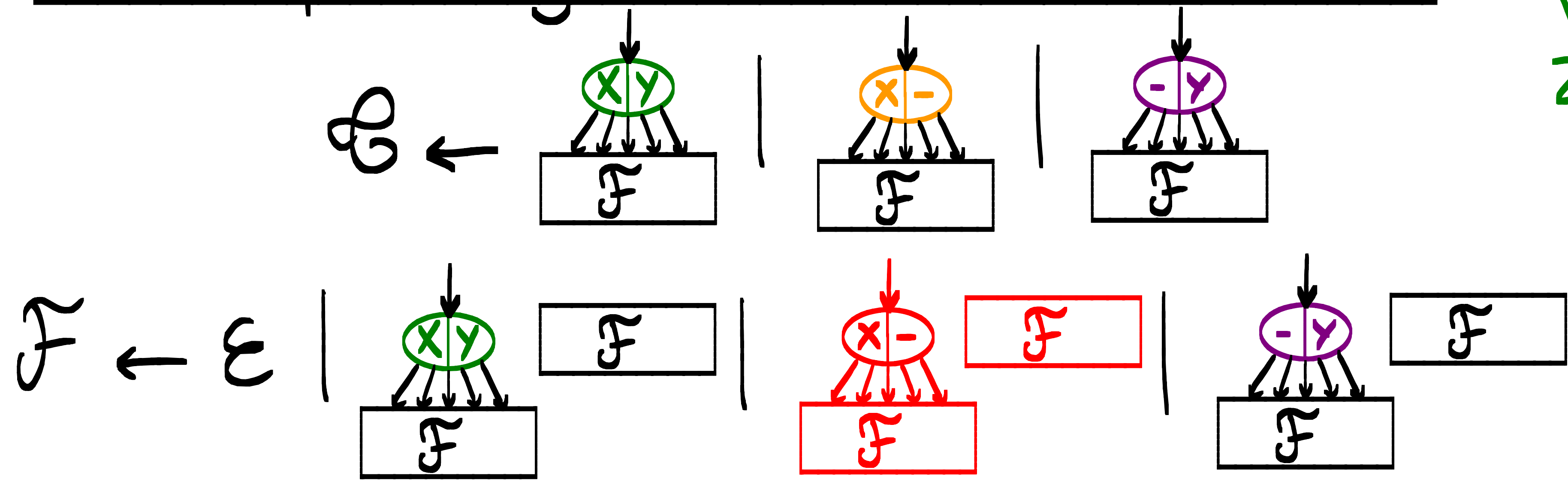


A GRAMMAR FOR ALIGNMENTS

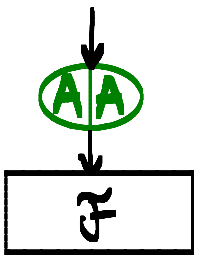
Strategy: Build a context-free grammar that generates every alignment exactly once

<u>An example of grammar that does not work:</u>

[Jiang, Wang, Zhang]



Ex:

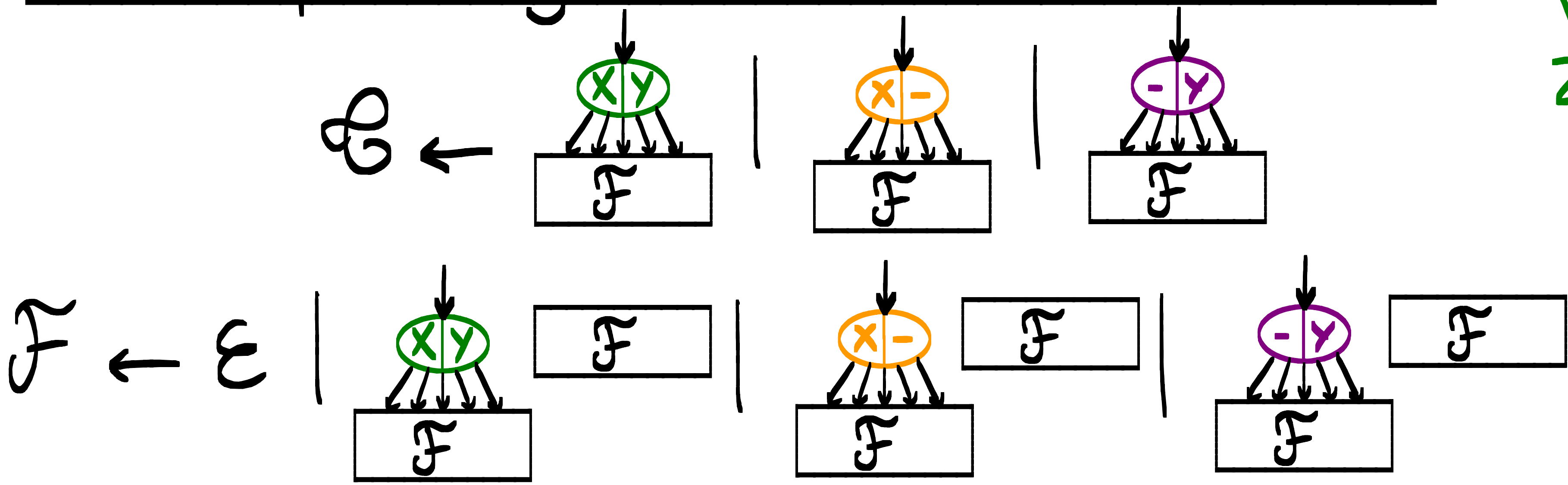


A GRAMMAR FOR ALIGNMENTS

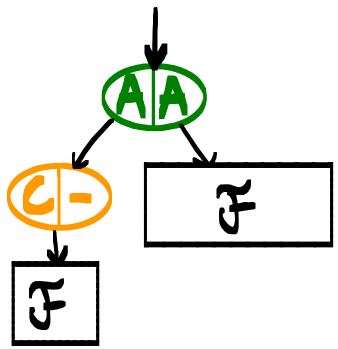
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:

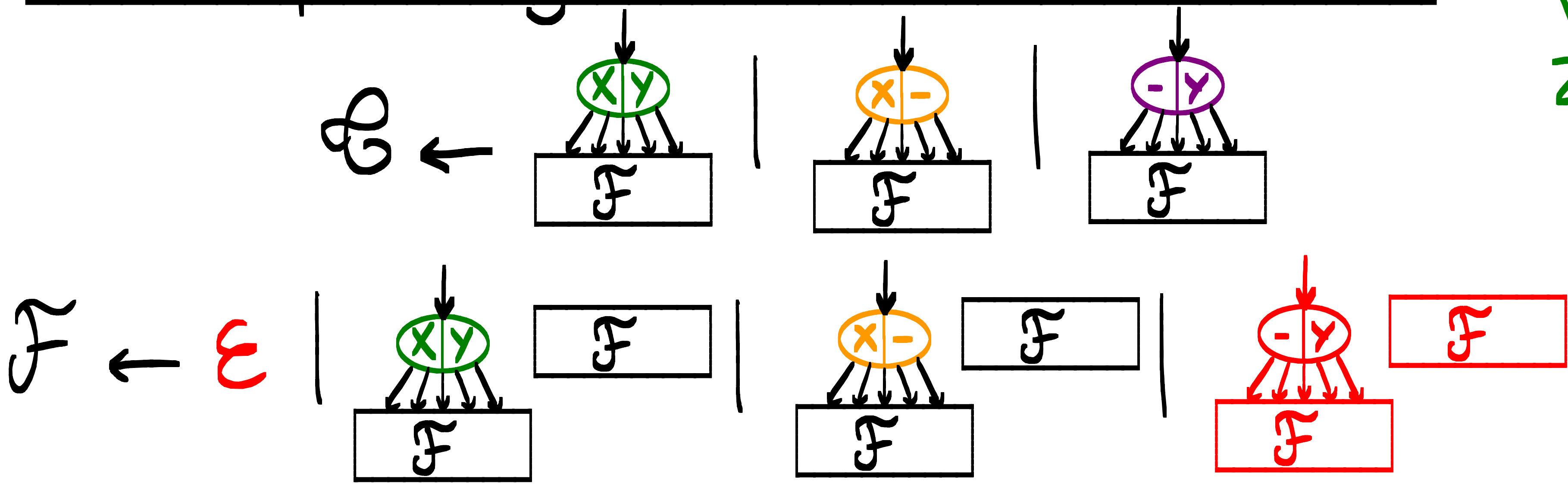


A GRAMMAR FOR ALIGNMENTS

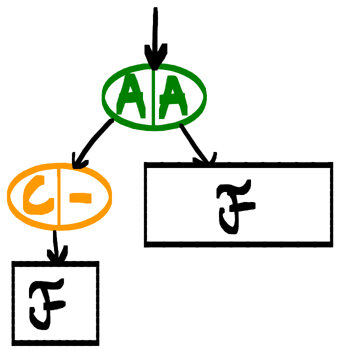
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:

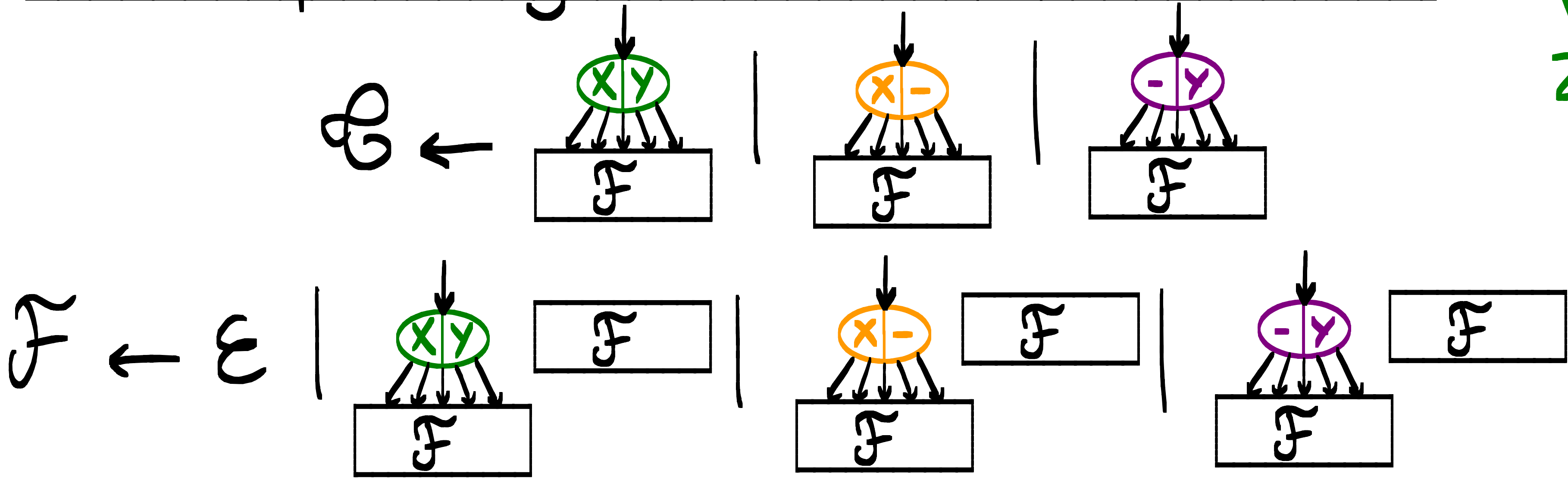


A GRAMMAR FOR ALIGNMENTS

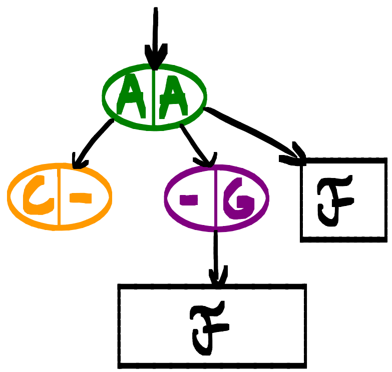
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:

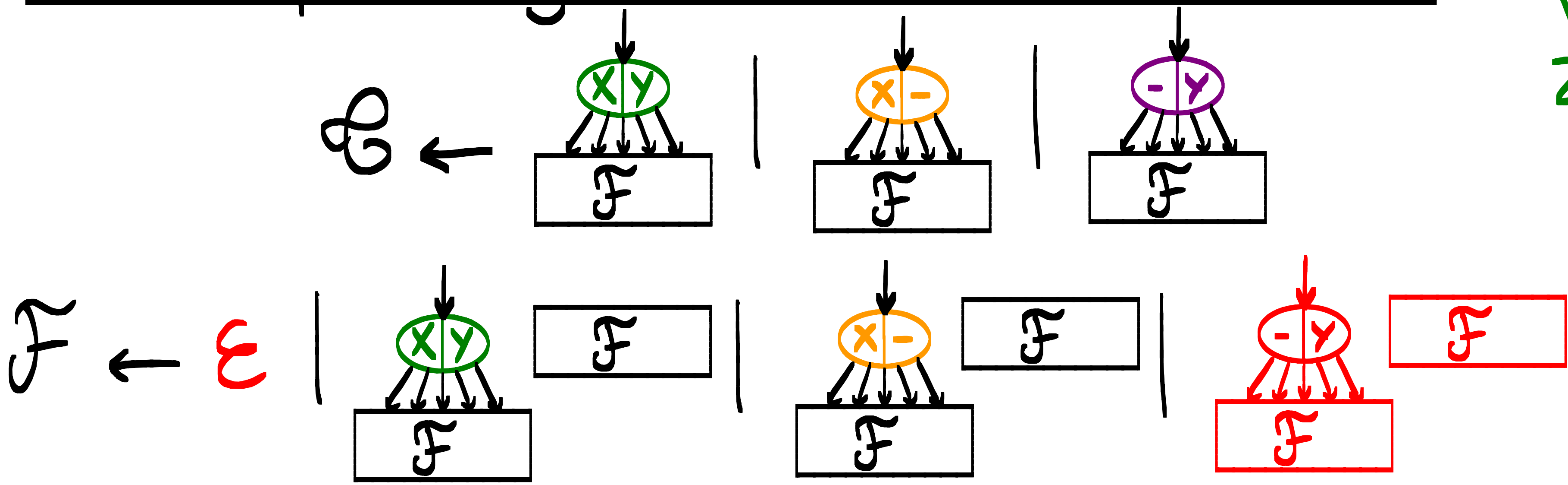


A GRAMMAR FOR ALIGNMENTS

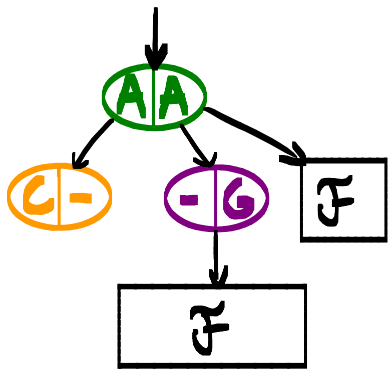
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:

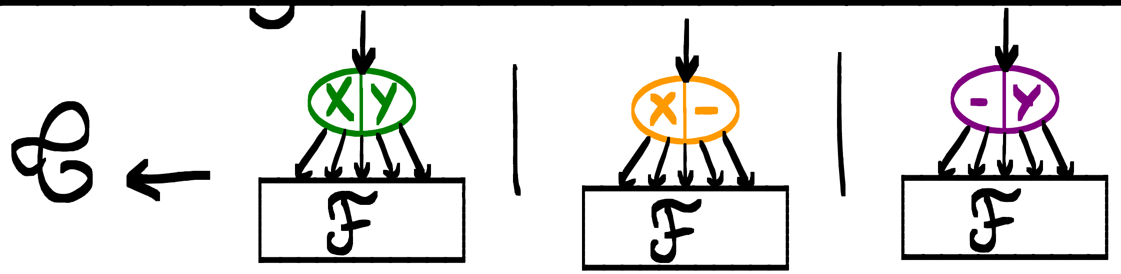


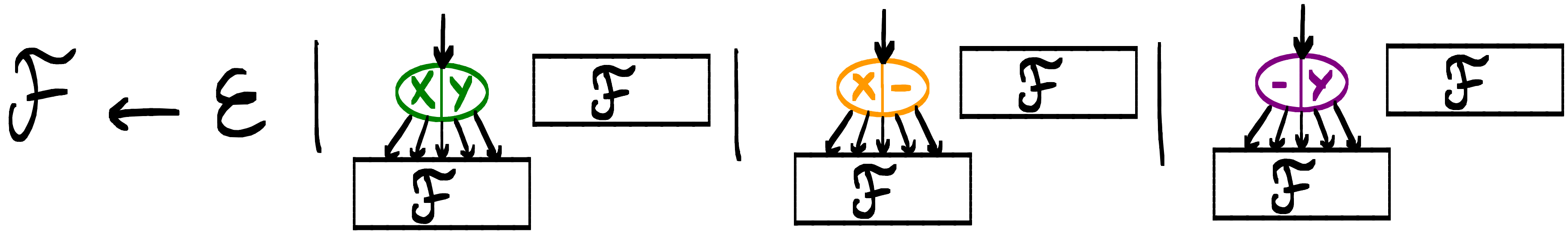
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

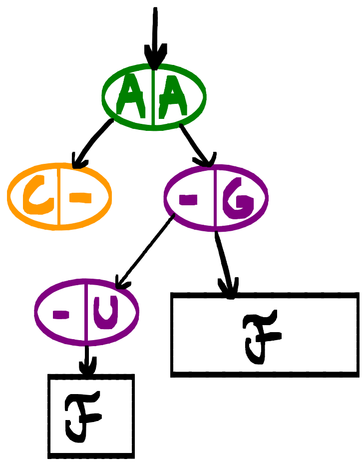
<u>An example of grammar that does not work:</u>

[Jiang, Wang, Zhang]





Ex:

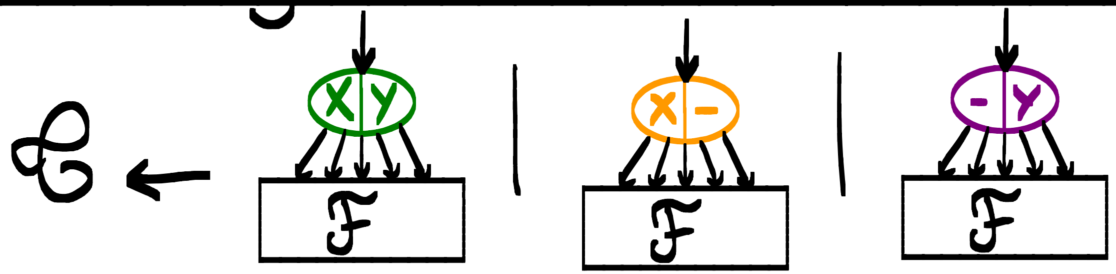


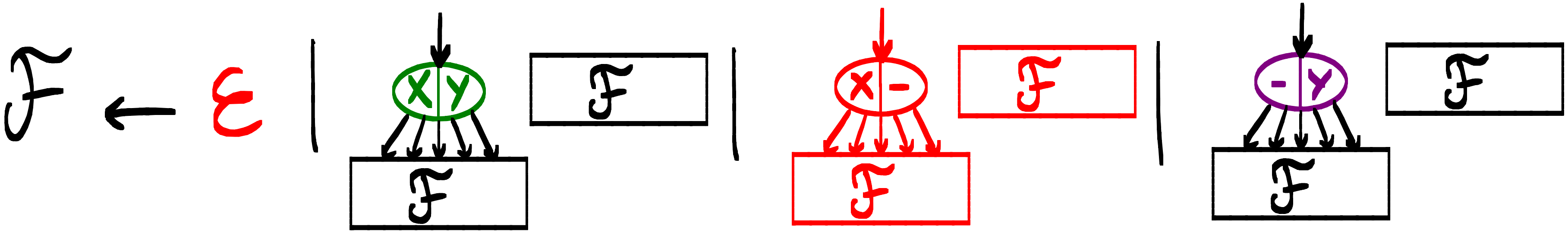
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

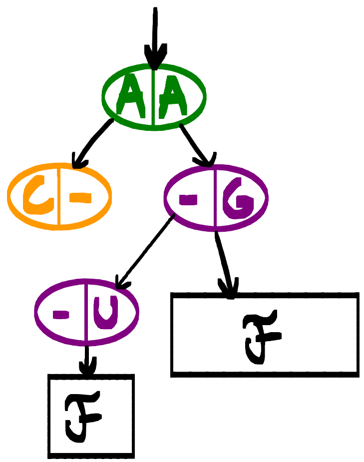
An example of grammar that does not work:

[Jiang, Wang, Zhang]





Ex:

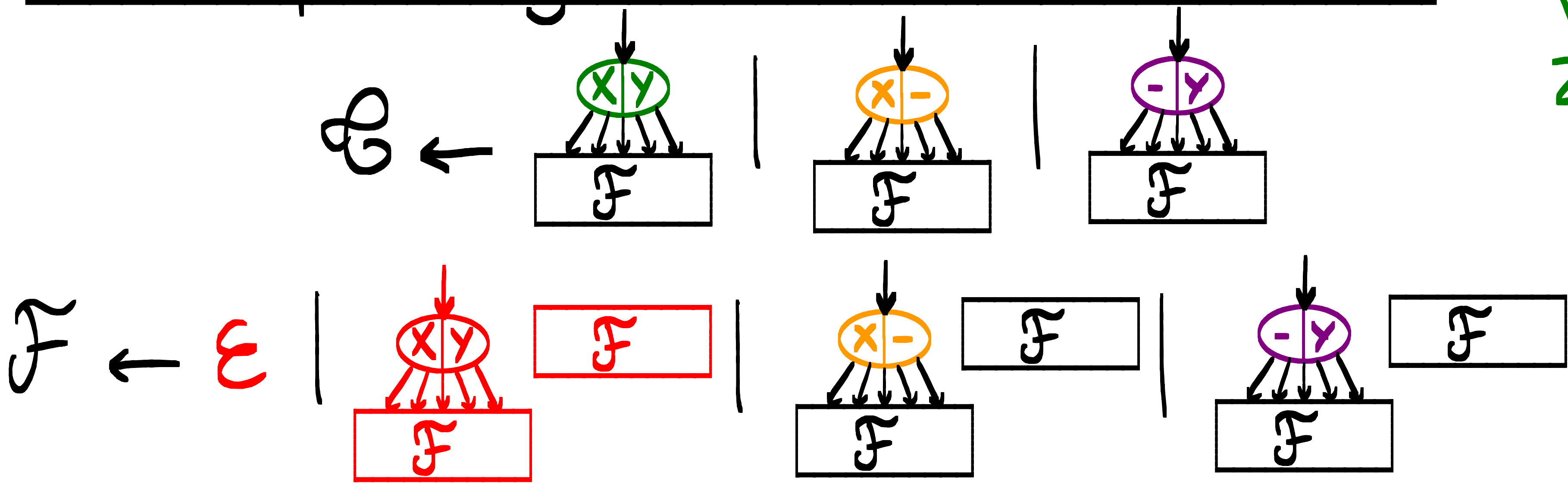


A GRAMMAR FOR ALIGNMENTS

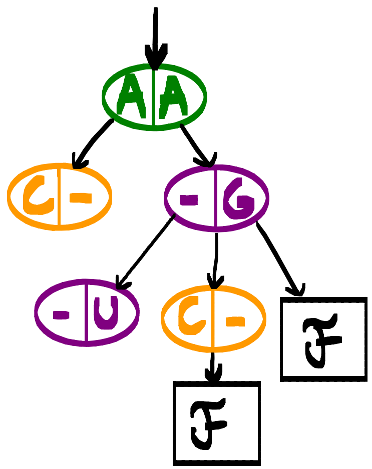
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:

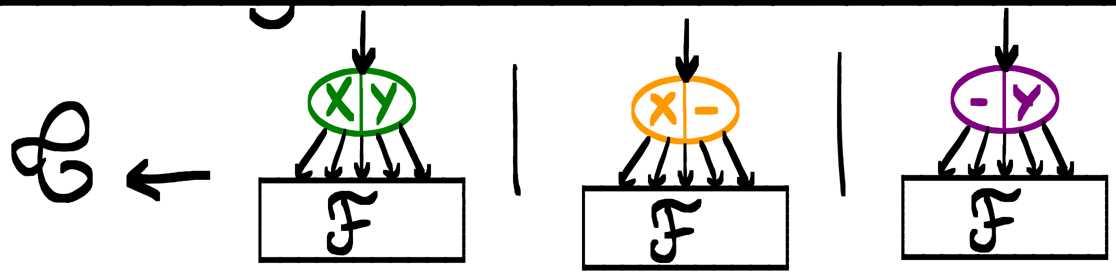


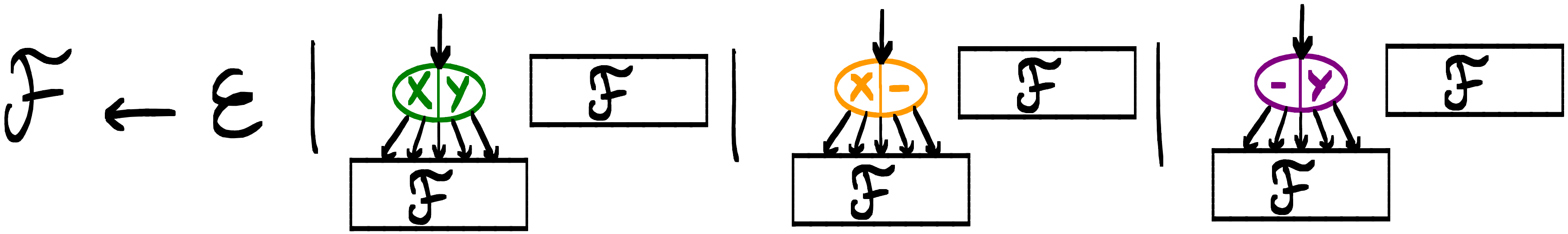
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

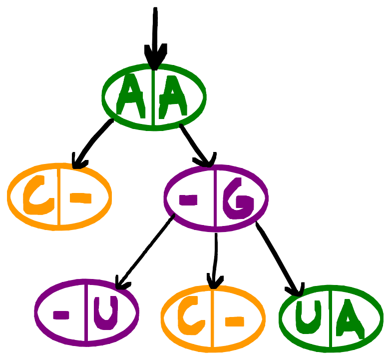
An example of grammar that does not work:

[Jiang, Wang, Zhang]





Ex:

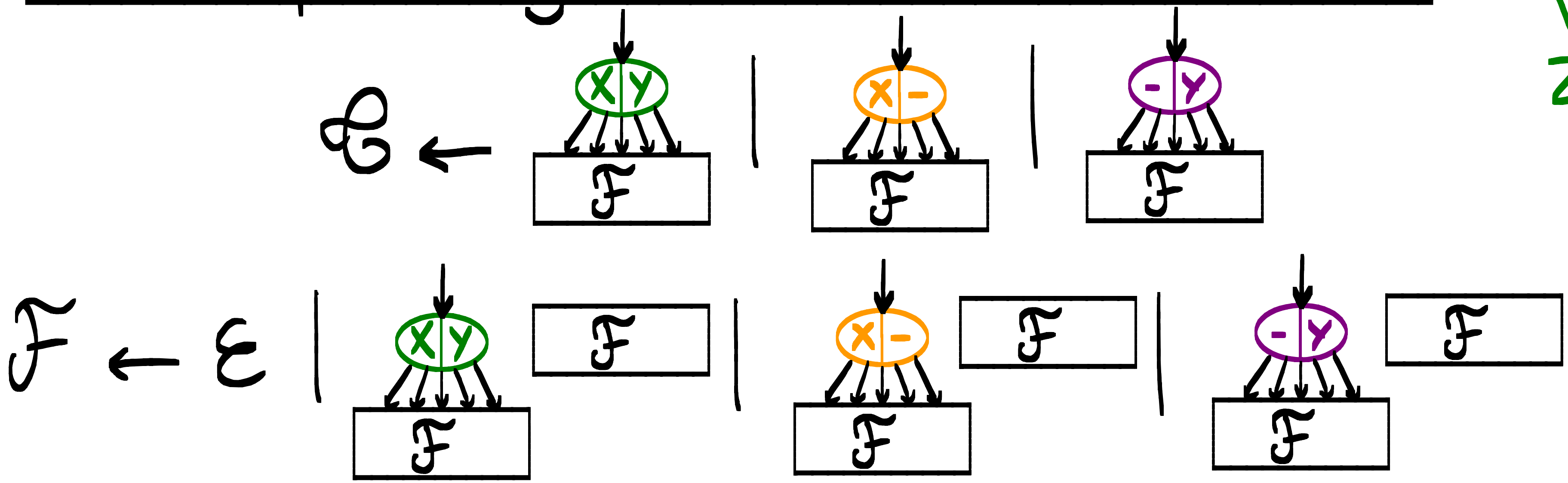


A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]

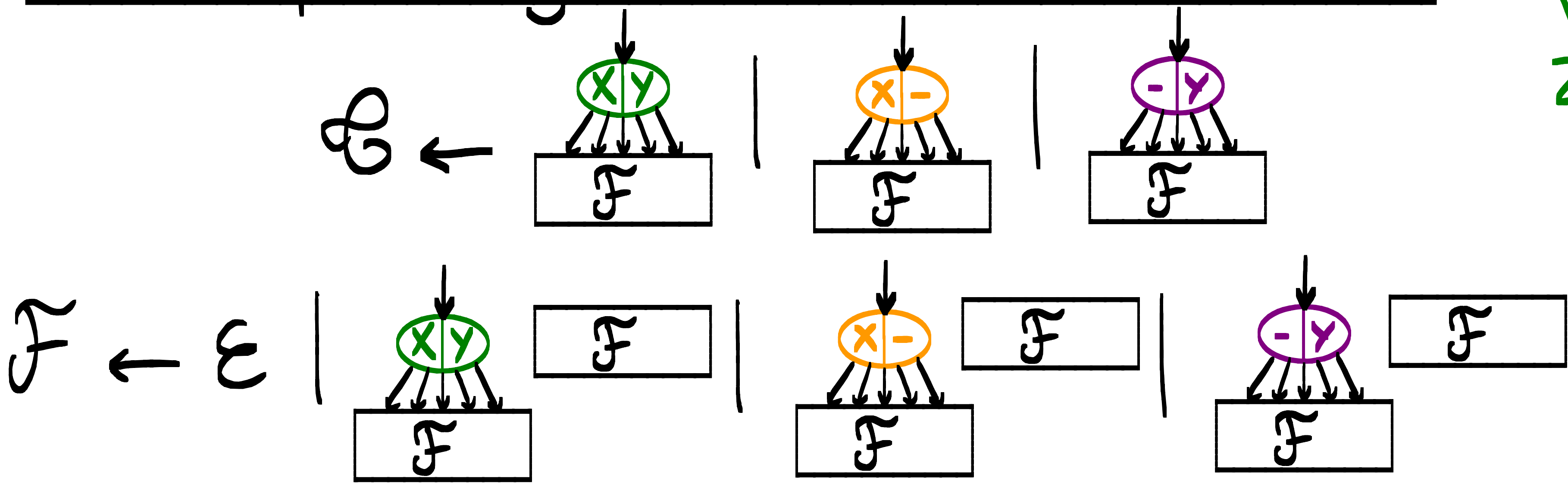


Ex:

A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work: [Jiang, Wang, Zhang]



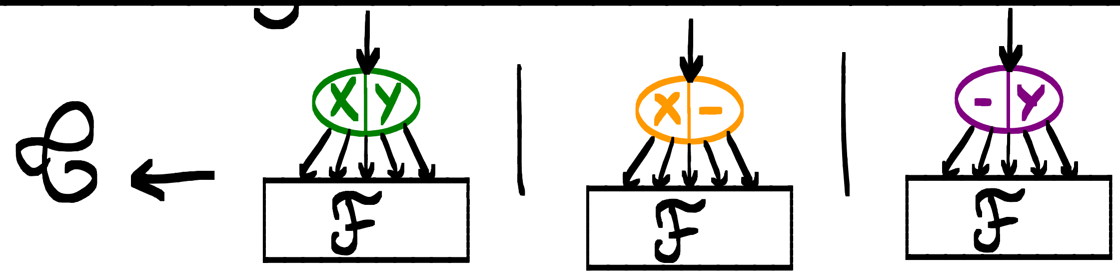
Ex:

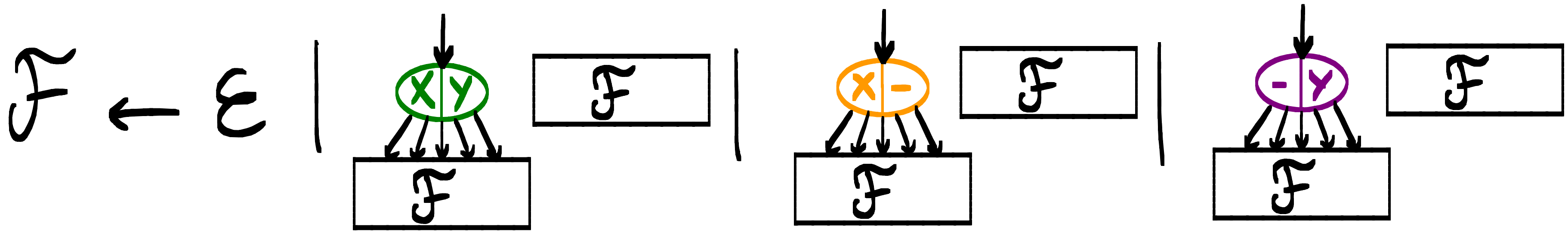
A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

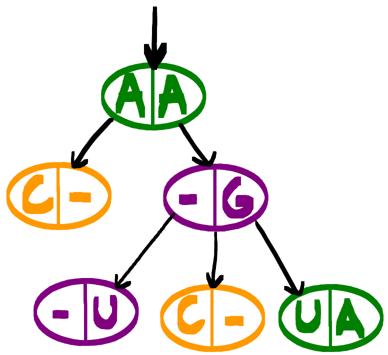
An example of grammar that does not work:

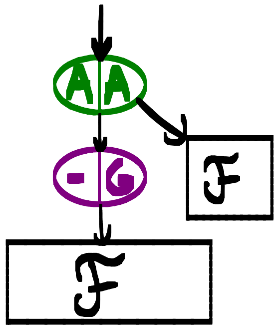
[Jiang, Wang, Zhang]





Ex:



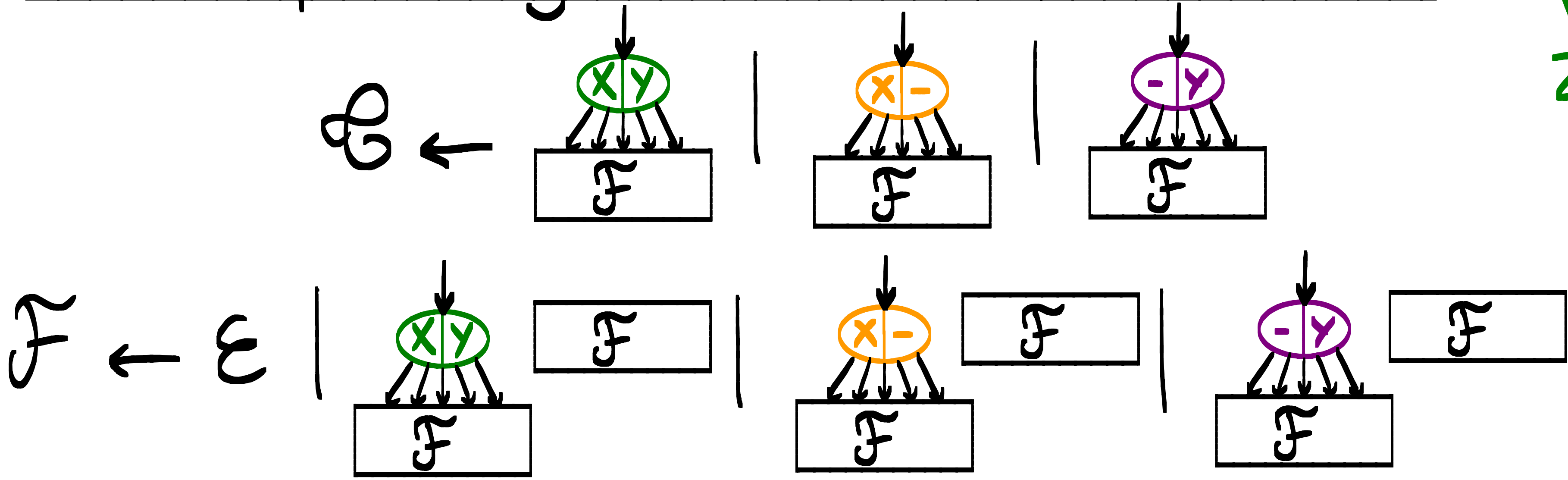


A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



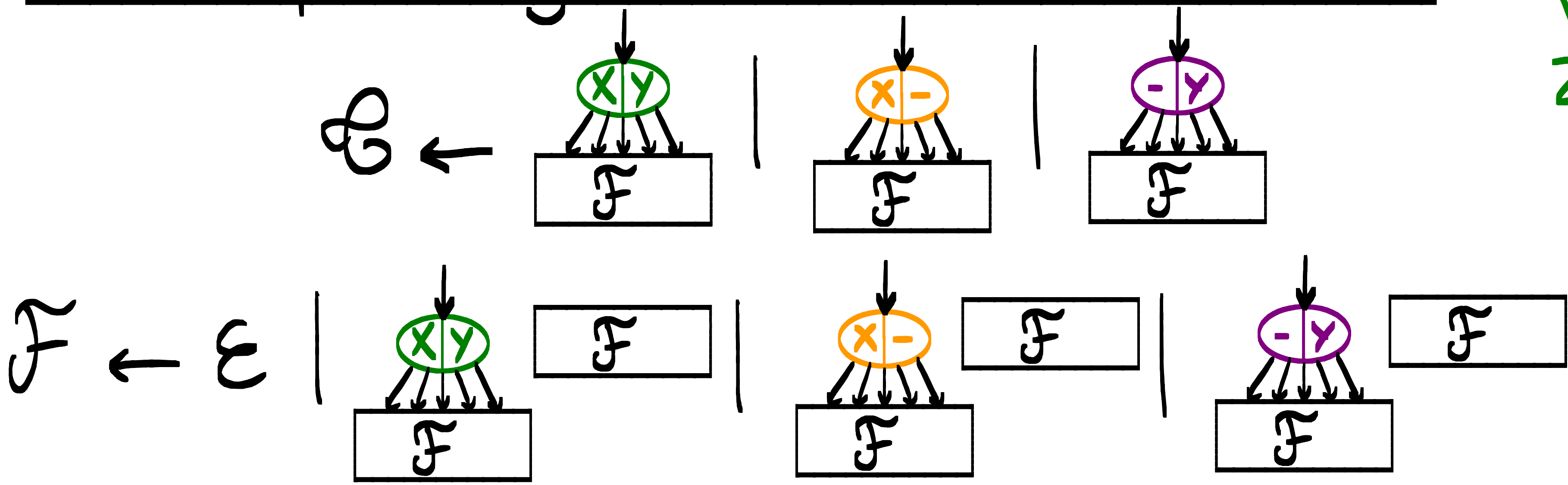
Ex:

A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



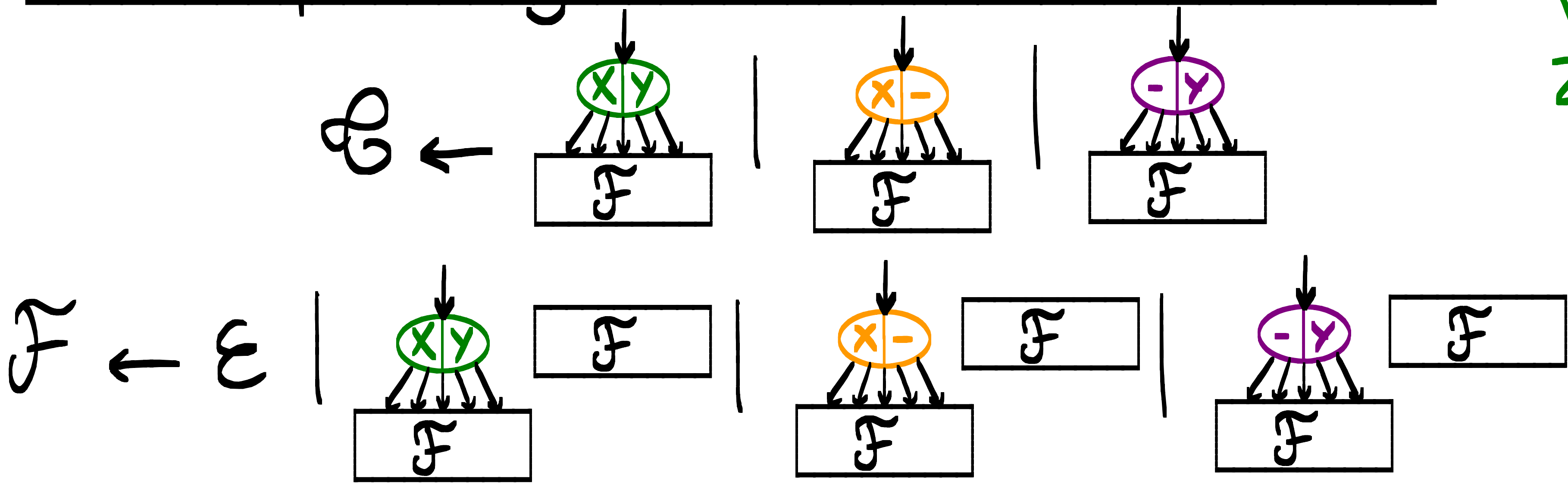
Ex:

A GRAMMAR FOR ALIGNMENTS

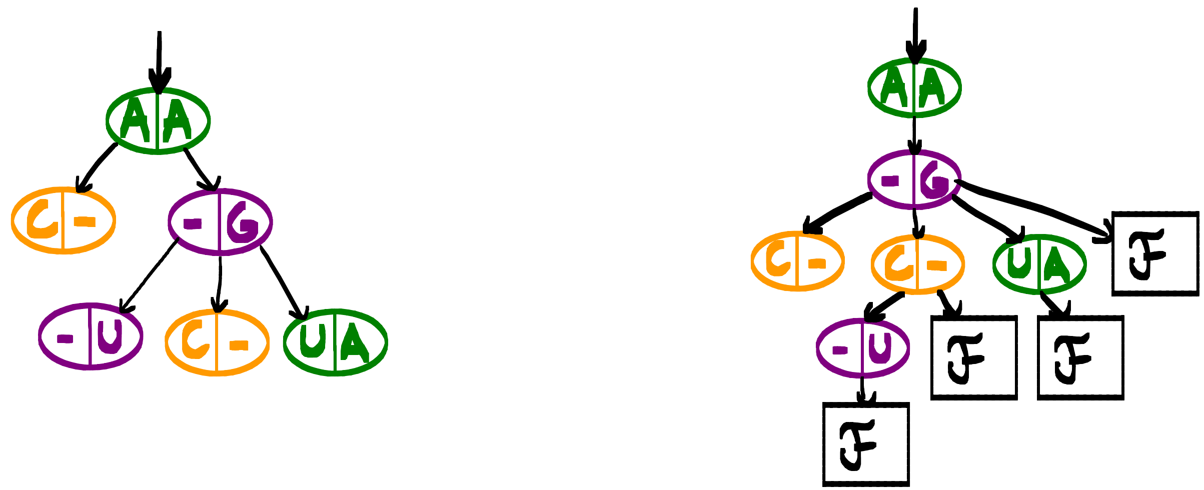
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



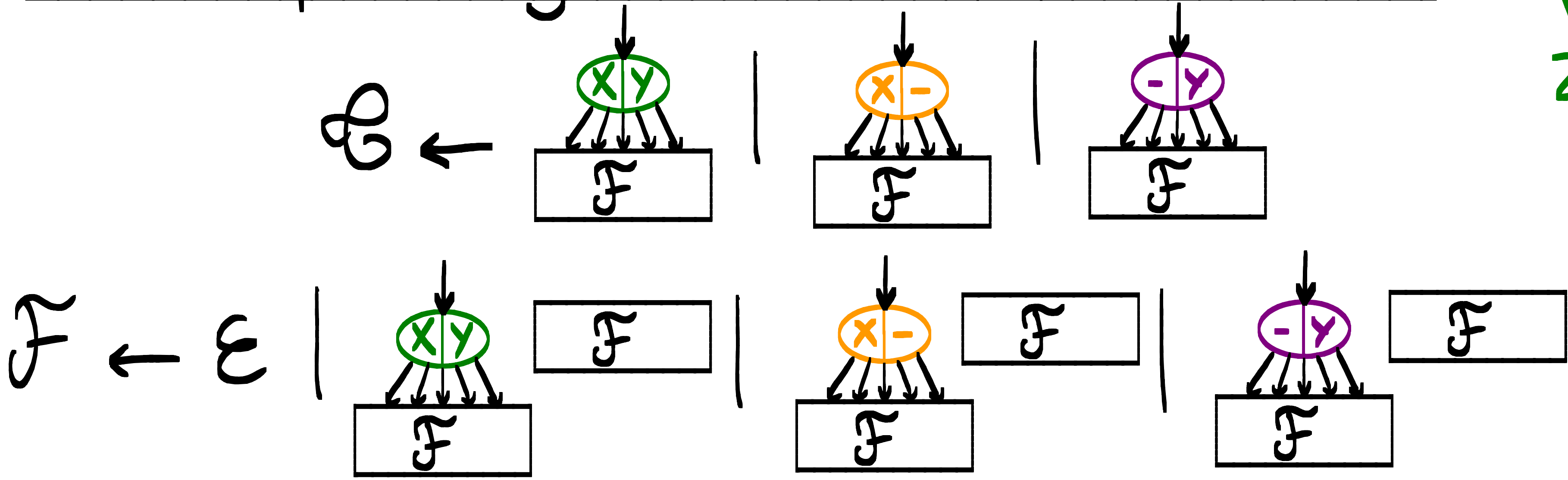
Ex:



A GRAMMAR FOR ALIGNMENTS

Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work: [Jiang, Wang, Zhang]



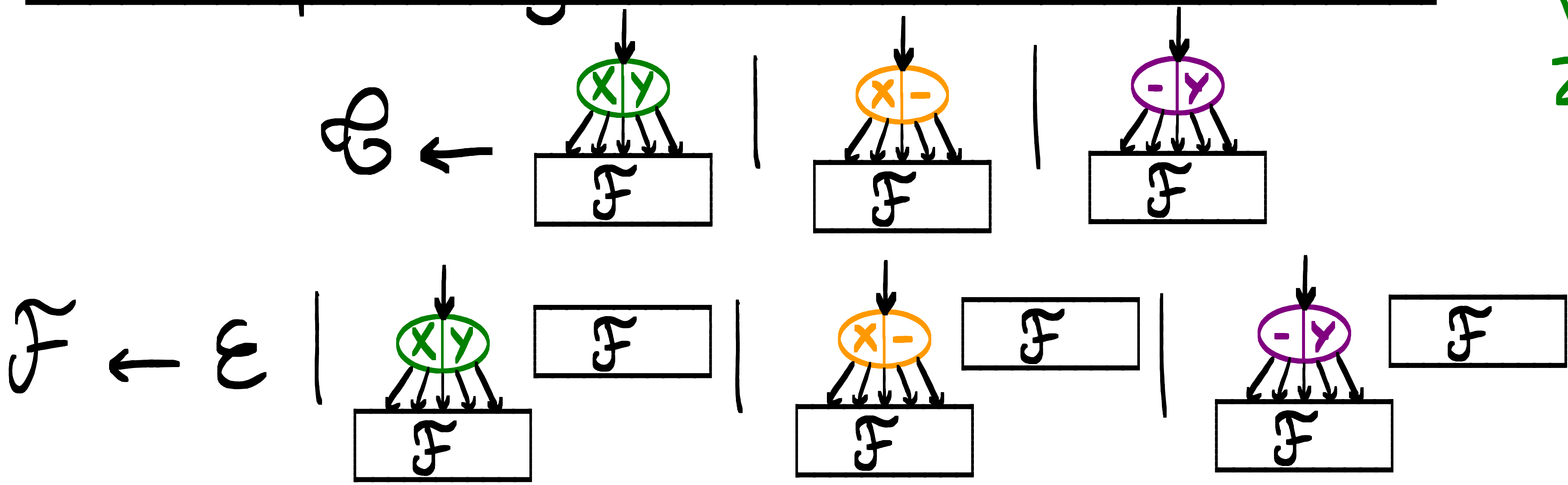
Ex:

A GRAMMAR FOR ALIGNMENTS

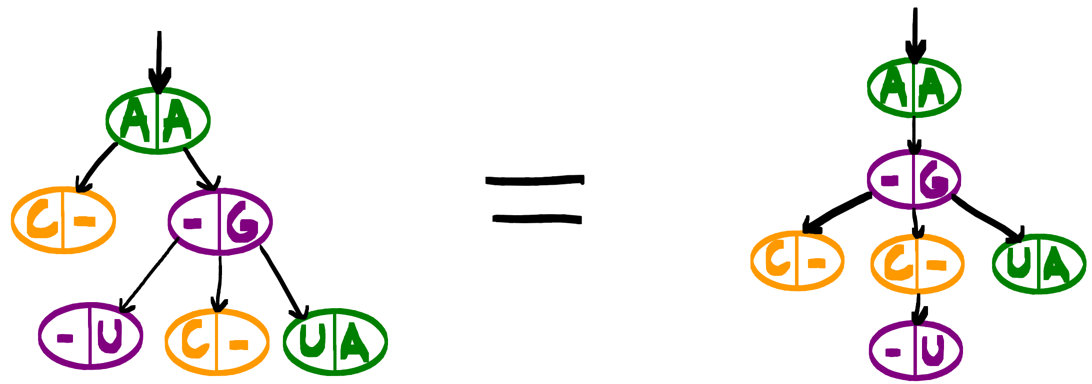
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:



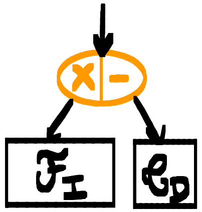
ambiguous grammar

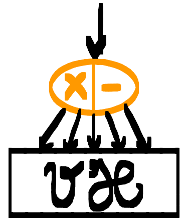
A GRAMMAR FOR ALIGNMENTS

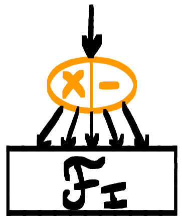
Theorem: The set of generated by the following grammar contains every tree alignment exactly once.

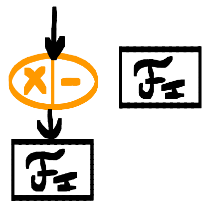
A GRAMMAR FOR ALIGNMENTS

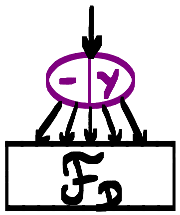
Our (complicated) non-ambiguous grammar:

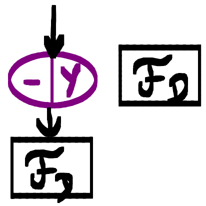
$\mathcal{A} \leftarrow \mathcal{V}^{\emptyset} \mid \mathcal{C}_I \mid \mathcal{C}_D \mid$ 

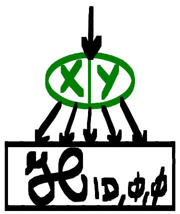
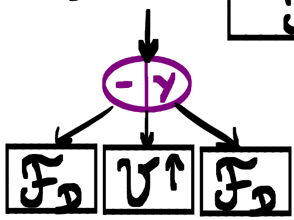
$\mathcal{V}^{\emptyset} \leftarrow \mathcal{V}^{\uparrow} \mid$ 

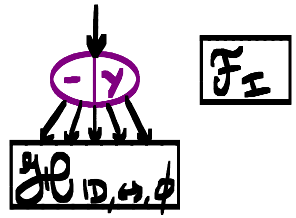
$\mathcal{C}_I \leftarrow$ 

$\mathcal{F}_I \leftarrow \varepsilon \mid$ 

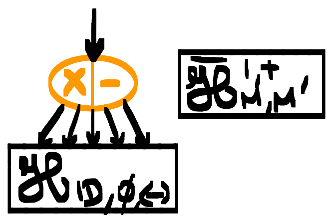
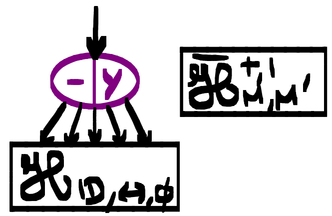
$\mathcal{C}_D \leftarrow$ 

$\mathcal{F}_D \leftarrow \varepsilon \mid$ 

$\mathcal{V}^{\uparrow} \leftarrow$  $\mid$ 

$\mathcal{VH} \leftarrow \boxed{\mathcal{F}_I}\boxed{\mathcal{VH}} \mid \boxed{\mathcal{V}^{\emptyset}}\boxed{\mathcal{F}_I} \mid$ 

For $\mathcal{V} \in \{\mathbb{D}, \mathcal{D}\}$, $(M, M') \in \{\emptyset, \rightarrow, \leftrightarrow\}^2$:

$\mathcal{B}_{\mathcal{V},M,M'} \leftarrow \varepsilon \mid \boxed{\mathcal{C}_I}\boxed{\mathcal{B}_{\mathcal{V},M,M'}} \mid \boxed{\mathcal{C}_D}\boxed{\mathcal{B}_{\mathcal{D},M,M'}} \mid \boxed{\mathcal{V}^{\emptyset}}\boxed{\overline{\mathcal{B}}_{M,M'}} \mid$  $\mid$ 

only if $(M,M') = (\emptyset,\emptyset)$

only if $\mathcal{V} \neq \mathcal{D}$ and $M \neq \leftrightarrow$

only if $M' \neq \leftrightarrow$

no room for $\overline{\mathcal{B}}^{i,j}_{M,M'}$...

APPLICATION 1: COUNTING.

a_n = number of tree alignments of size n

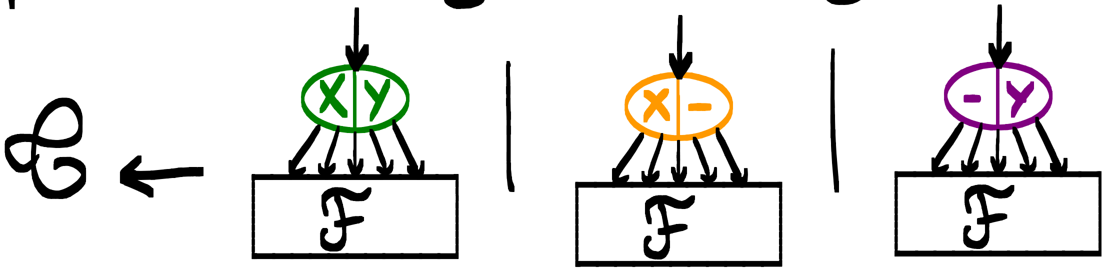
Generating function: $A(z) = \sum_{n \geq 0} a_n z^n$

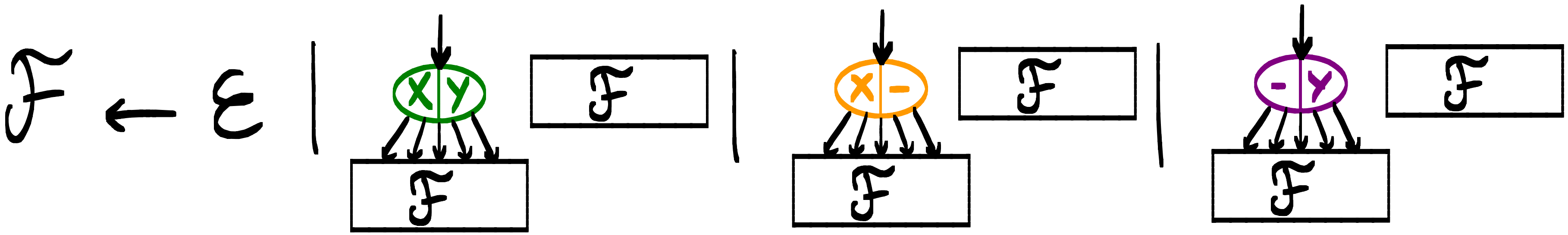
APPLICATION 1: COUNTING.

a_n = number of tree alignments of size n

Generating function: $A(z) = \sum_{n \geq 0} a_n z^n$

The principle on Jiang et al.'s grammar:



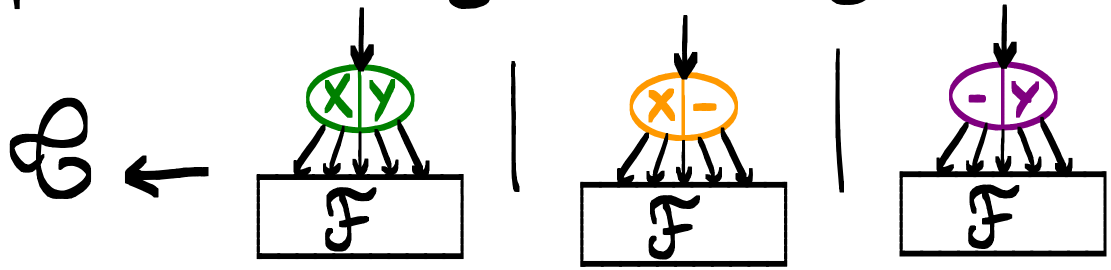


APPLICATION 1: COUNTING.

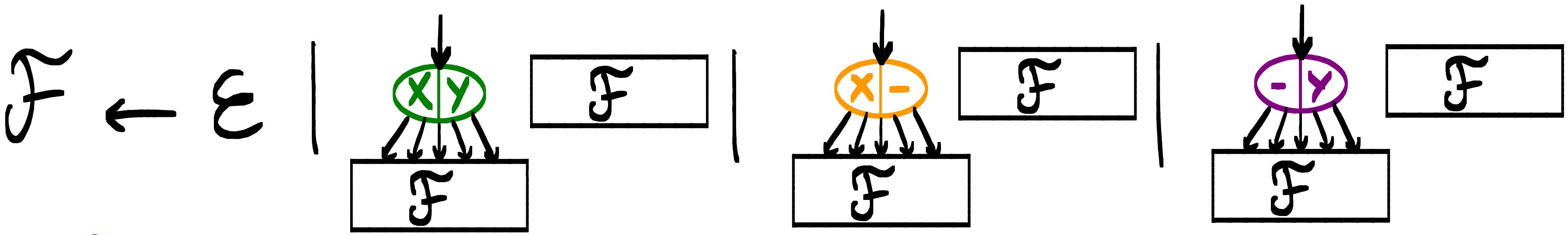
a_n = number of tree alignments of size n

Generating function: $A(z) = \sum_{n \geq 0} a_n z^n$

The principle on Jiang et al.'s grammar:



$$T(z) = F(z) + F(z) + F(z)$$



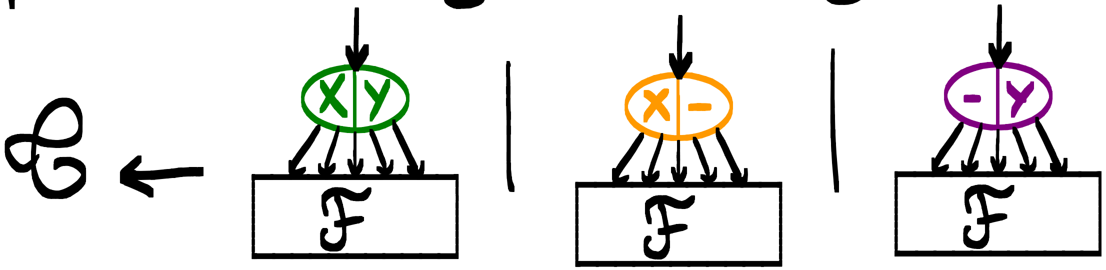
$$F(z) = 1 + F(z) \times F(z) + F(z) \times F(z) + F(z) \times F(z)$$

APPLICATION 1: COUNTING.

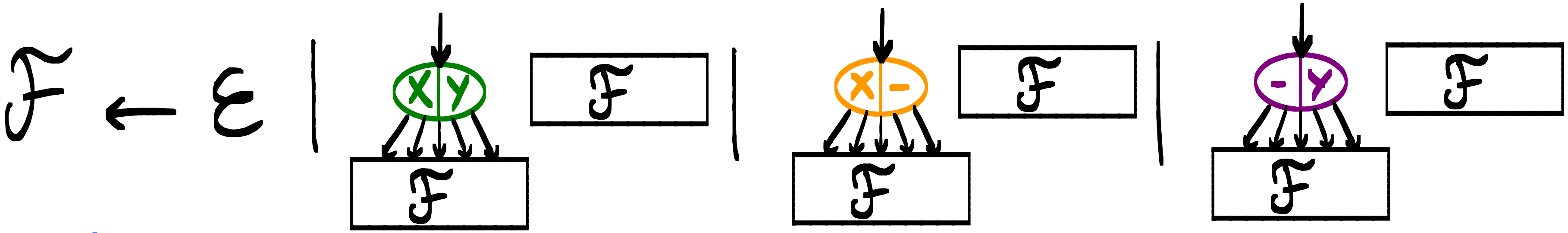
a_n = number of tree alignments of size n

Generating function: $A(z) = \sum_{n \geq 0} a_n z^n$

The principle on Jiang et al.'s grammar:



$$T(z) = F(z) + F(z) + F(z)$$



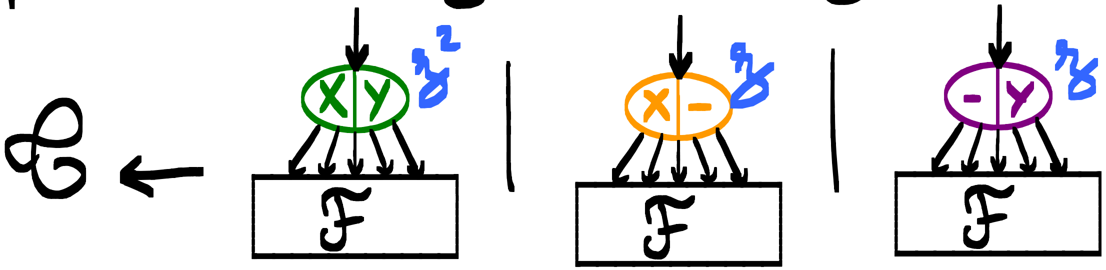
$$F(z) = 1 + F(z)^2 + F(z)^2 + F(z)^2$$

APPLICATION 1: COUNTING.

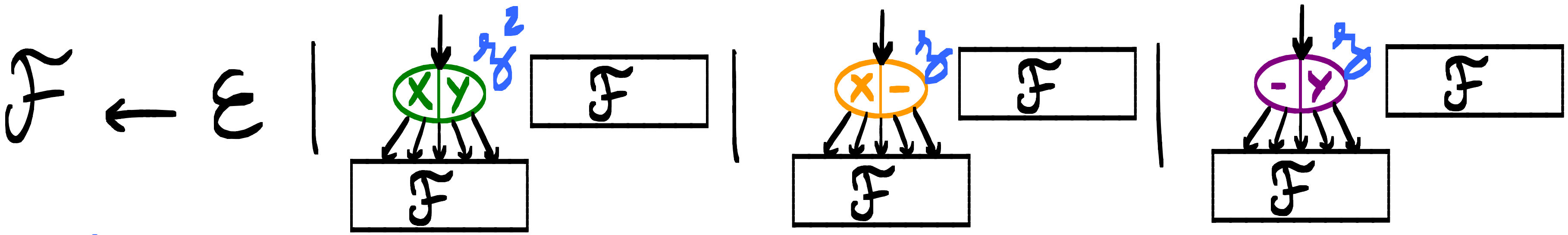
a_n = number of tree alignments of size n

Generating function: $A(z) = \sum_{n \geq 0} a_n z^n$

The principle on Jiang et al.'s grammar:



$$T(z) = F(z) + F(z) + F(z)$$



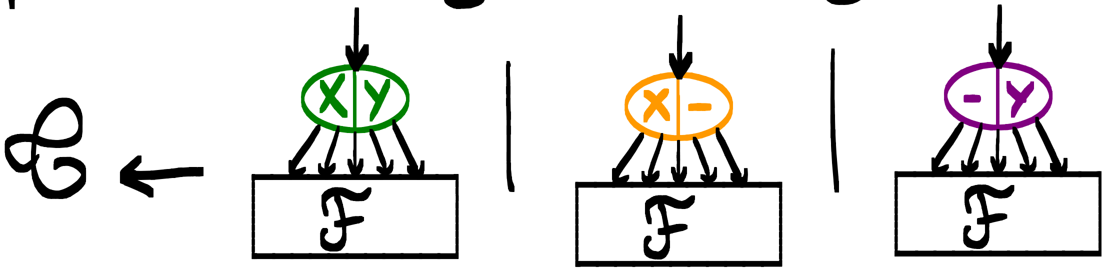
$$F(z) = 1 + F(z)^2 + F(z)^2 + F(z)^2$$

APPLICATION 1: COUNTING.

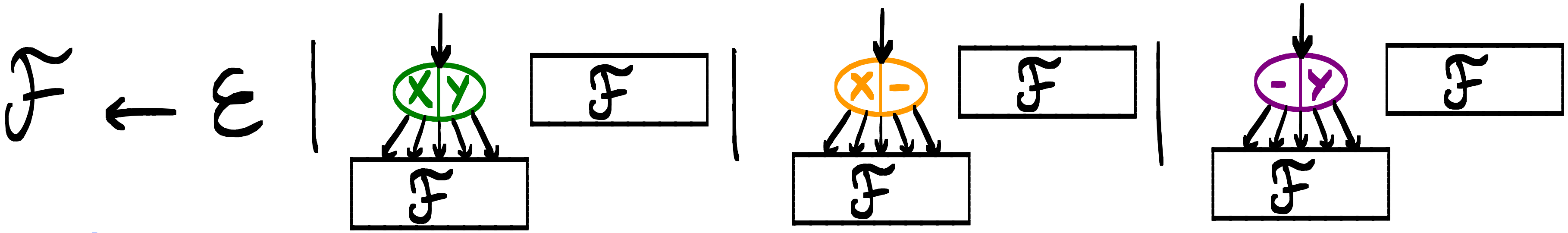
a_n = number of tree alignments of size n

Generating function: $A(z) = \sum_{n \geq 0} a_n z^n$

The principle on Jiang et al.'s grammar:



$T(z) = z^2 \times F(z) + z \times F(z) + z \times F(z)$



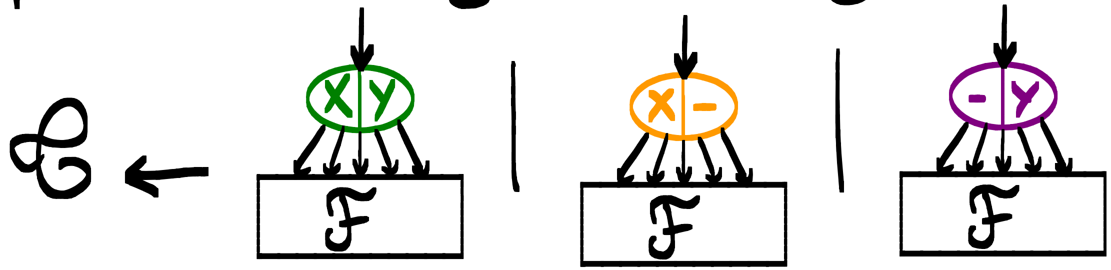
$F(z) = 1 + z^2 \times F(z)^2 + z \times F(z)^2 + z \times F(z)^2$

APPLICATION 1: COUNTING.

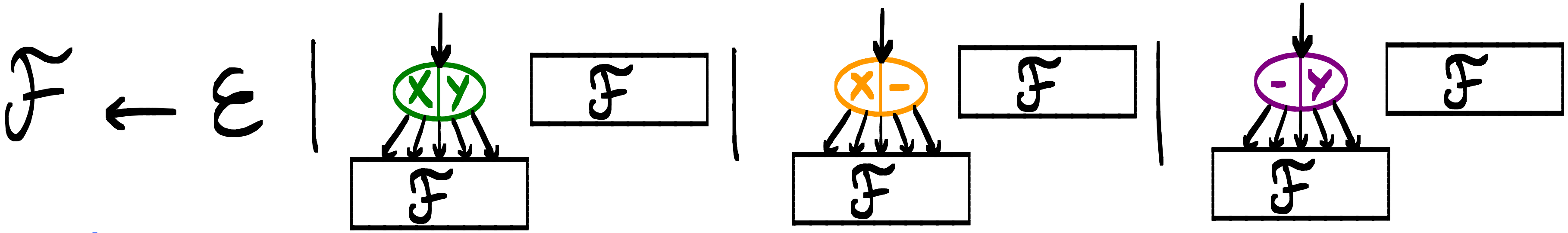
$a_{n,k}$ = number of tree alignments of size n and k matches

Generating function: $A(z,u) = \sum_{n \geq 0} a_{n,k}\, z^n u^k$

The principle on Jiang et al.'s grammar:



$$T(z,u) = u \times z^2 F(z,u) + z \times F(z,u) + z \times F(z,u)$$



$$F(z,u) = 1 + u \times z^2 \times F(z,u)^2 + z \times F(z,u)^2 + z \times F(z,u)^2$$

APPLICATION 1: COUNTING.

Theorem: The generating function $A(z,u)$ of tree alignments satisfies

$$A(z,u) = \left(z^2 + z - uz^2 + \frac{z}{\sqrt{1-4z}}\right) \times B(z,u)$$

where

$$\left(uzC(z)^2 - z^2C(z)^2 + 2z\right)B(z,u)^2 + \left(z^2C^4(z) - 2zC(z)^2 - 1\right)B(z,u) + C^2(z) = 0$$

and

$$C(z) = \frac{1-\sqrt{1-4z}}{2z}$$ Catalan generating function

SOME STATISTICAL PROPERTIES

<u>Theorem</u> There are on average

$C \times 1.5^n$ alignments

between two random trees of cumulative size n

where $C = 0.299\ldots$

<u>Corollary</u>: A same alignment was repeated

$\sim 0.875 \times 1.412^n$ times on

average in the previous

ambiguous grammar.

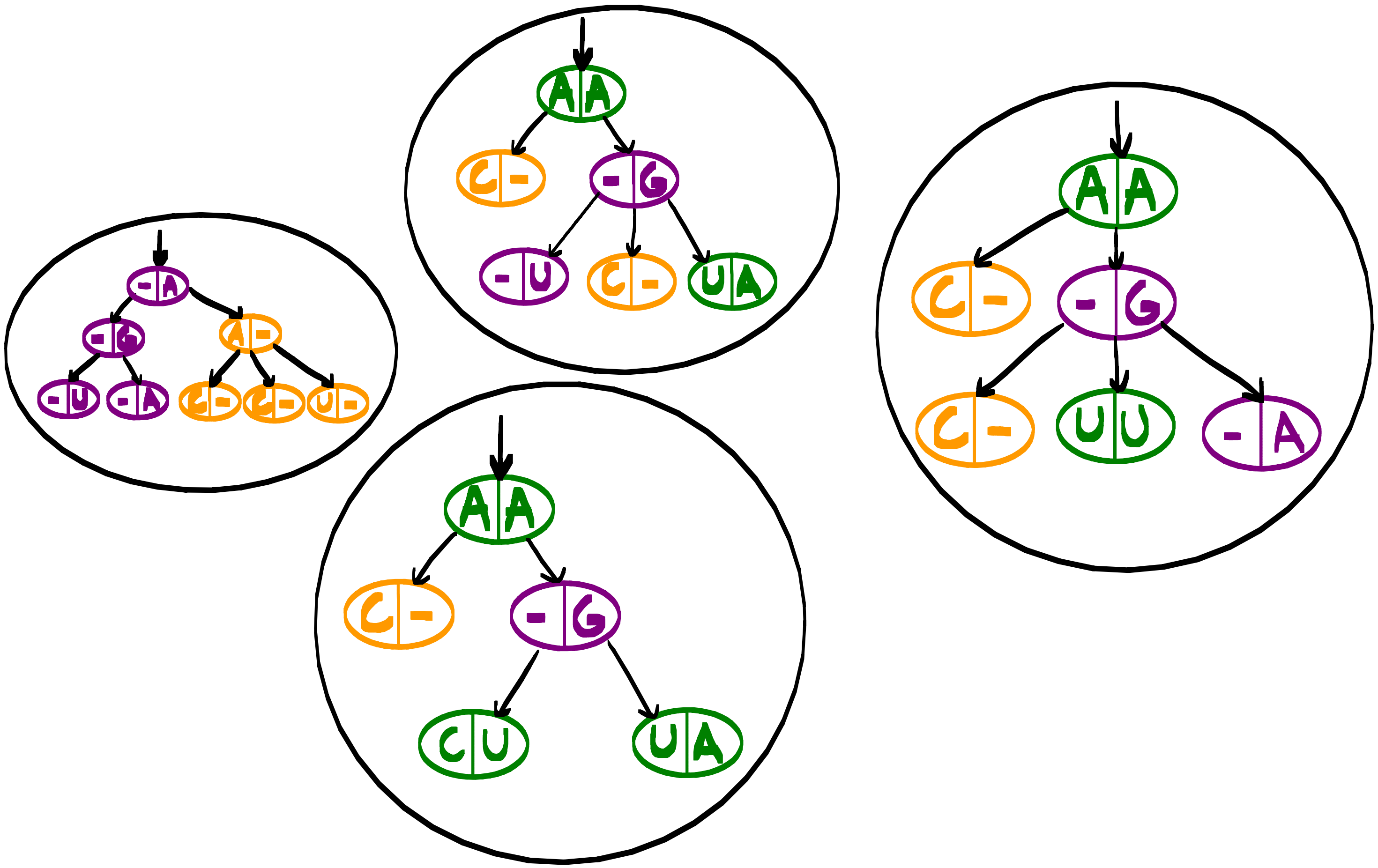
APPLICATION 2 - SAMPLING

Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

probability of an alignment A

$$\propto e^{-\frac{cost(A)}{K}}$$

(Gibbs-Boltzmann distribution)



APPLICATION 2. SAMPLING

Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

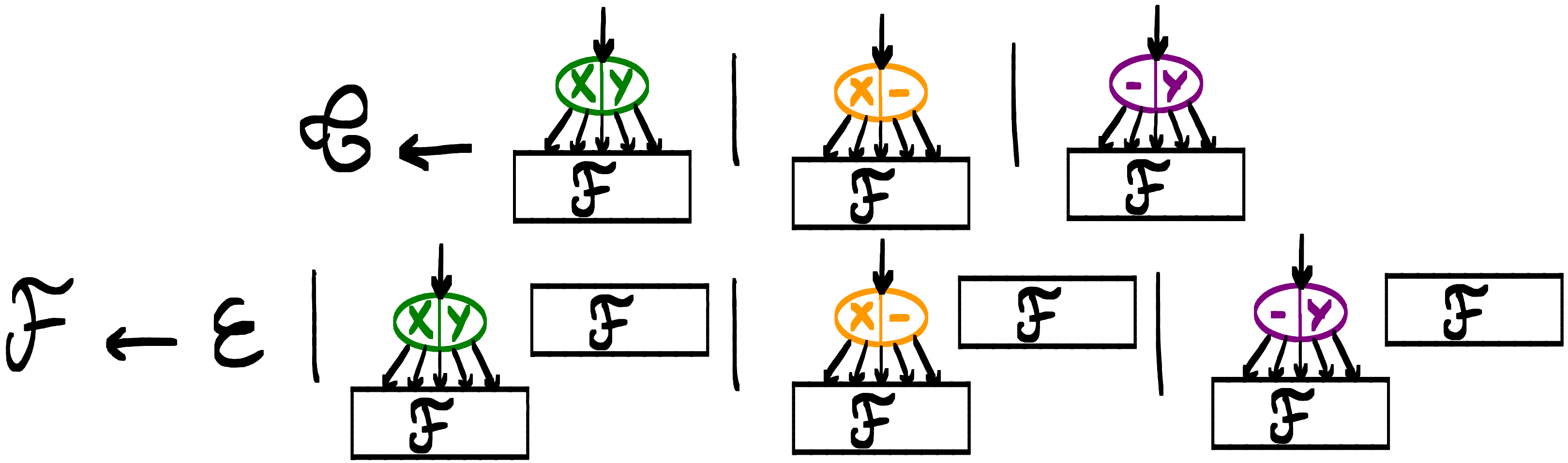
Strategy:

→ Filter the grammar to obtain a new grammar that only generates alignments between two fixed trees S and T

→ Use dynamic programming.

GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

The principle on Jiang et al.'s grammar:



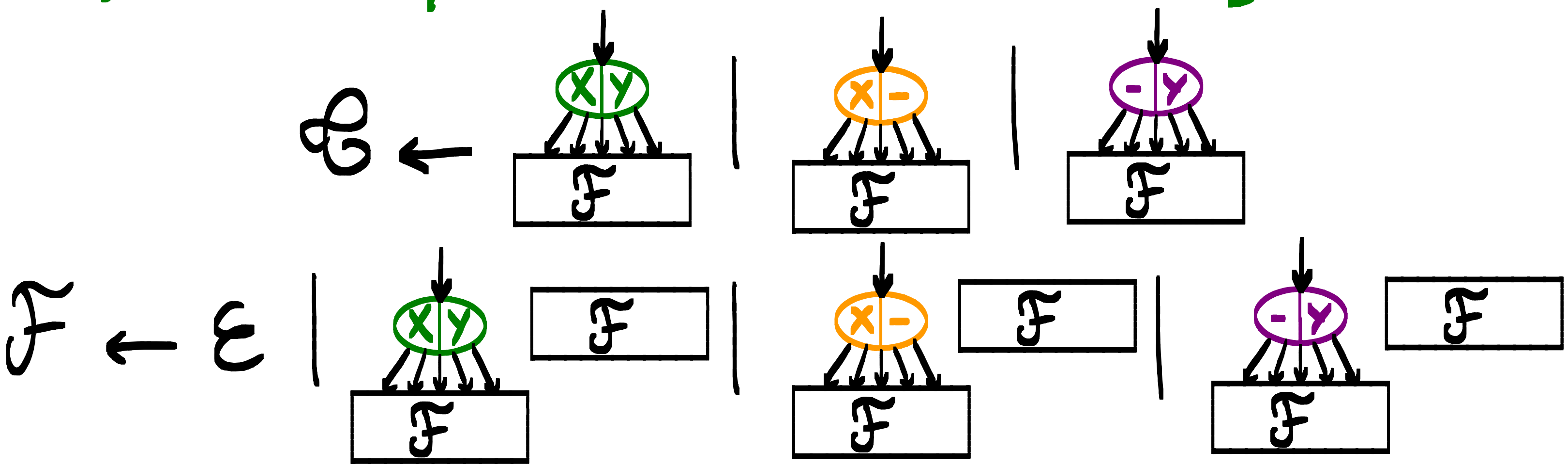
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

The principle on Jiang et al.'s grammar:

We fix two trees S and T.

Let F be a subforest of S and G a subforest of T

$J[F,G] = \{\text{alignments between } F \text{ and } G\}$



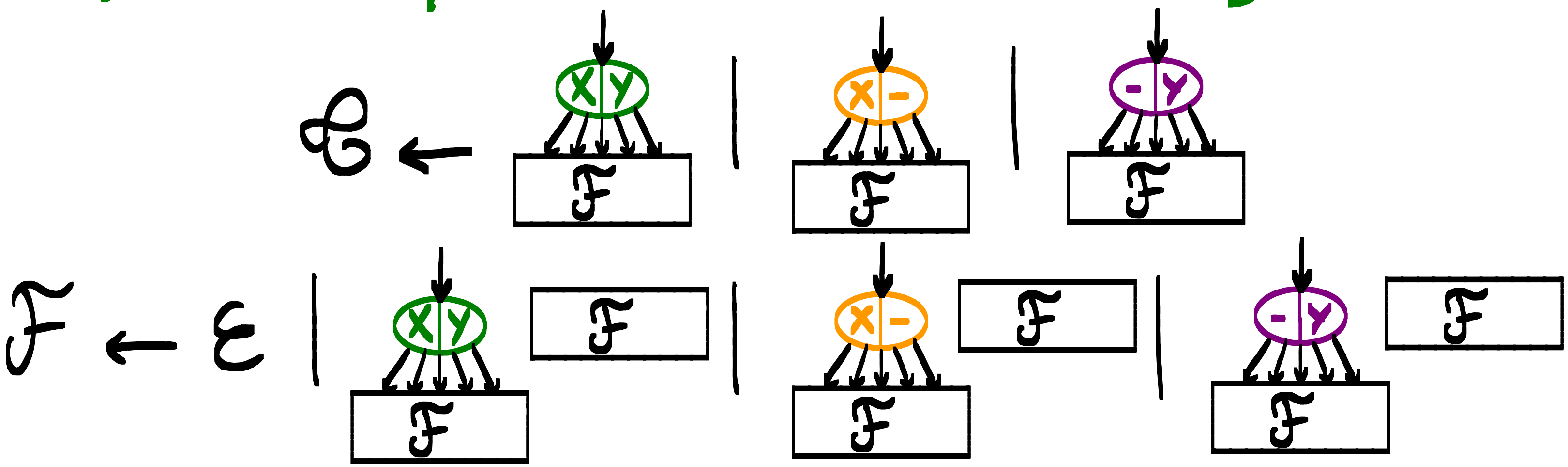
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

The principle on Jiang et al.'s grammar:

We fix two trees S and T.

Let F be a subforest of S and G a subforest of T

$J[F,G] = \{\text{alignments between } F \text{ and } G\}$



If $F = \emptyset$ and $G = \emptyset$, then

$$J[F,G] \leftarrow \varepsilon$$

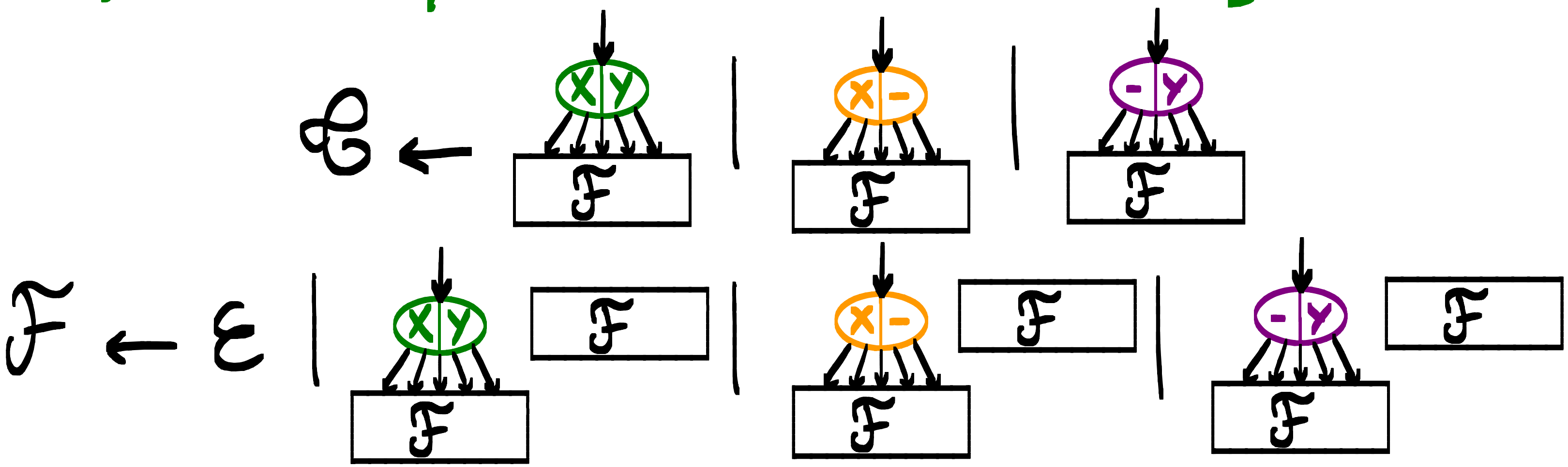
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

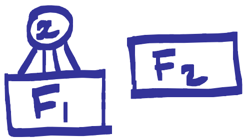
The principle on Jiang et al.'s grammar:

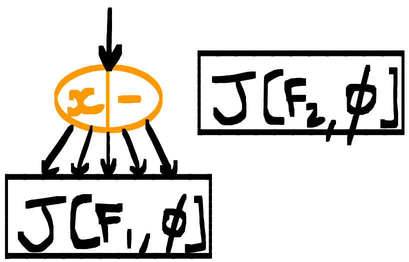
We fix two trees S and T.

Let F be a subforest of S and G a subforest of T

$J[F,G] = \{\text{alignments between } F \text{ and } G\}$



If $F =$  and $G = \emptyset$, then

$J[F,G] \leftarrow$ 

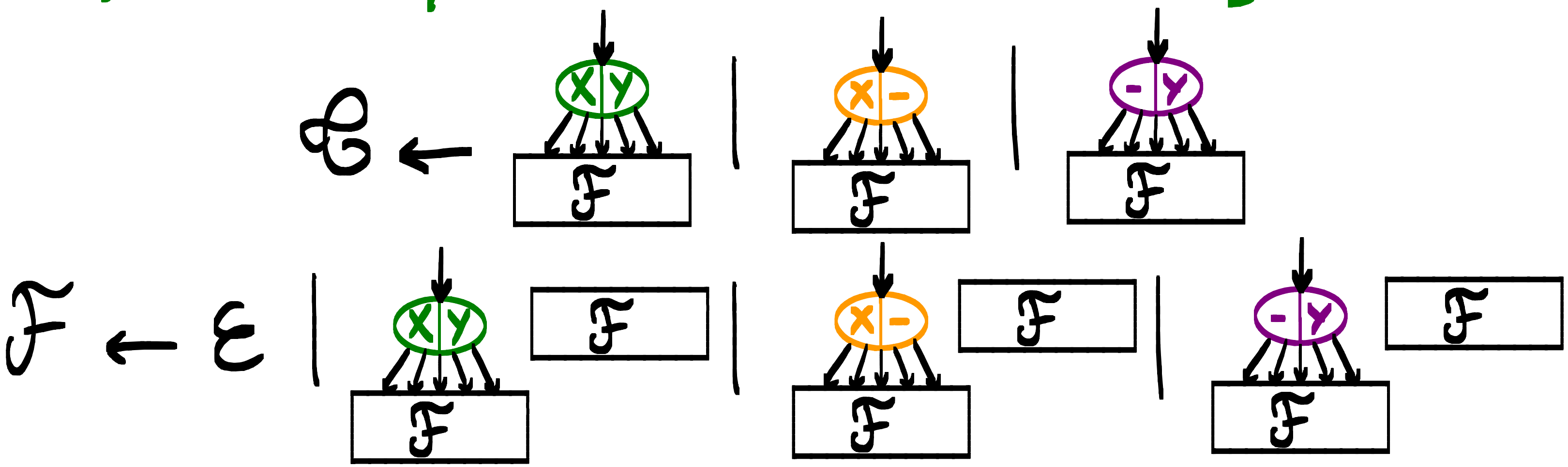
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

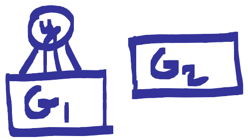
The principle on Jiang et al.'s grammar:

We fix two trees S and T.

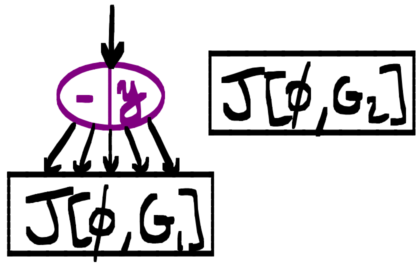
Let F be a subforest of S and G a subforest of T

$J[F,G] = \{$alignments between F and $G\}$



If $F = \emptyset$ and $G =$ 

, then

$J[F,G] \leftarrow$ 

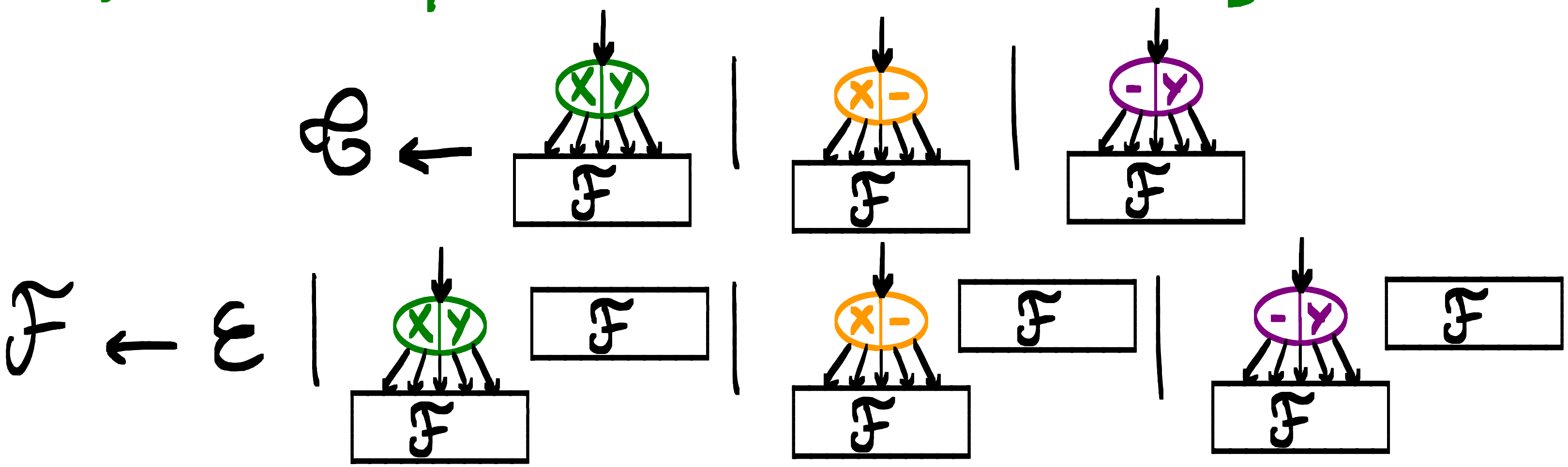
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

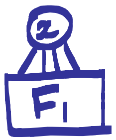
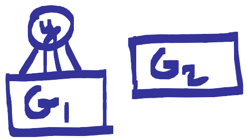
The principle on Jiang et al.'s grammar:

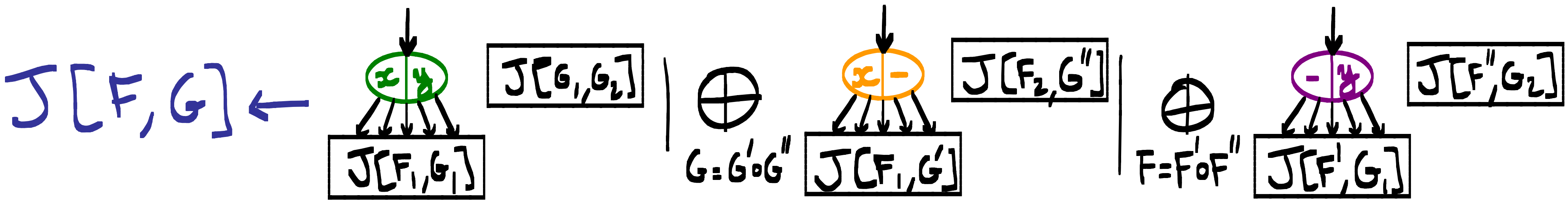
We fix two trees S and T.

Let F be a subforest of S and G a subforest of T

$J[F,G]$ = {alignments between F and G}

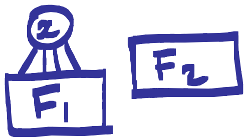


If F =  and G = , then

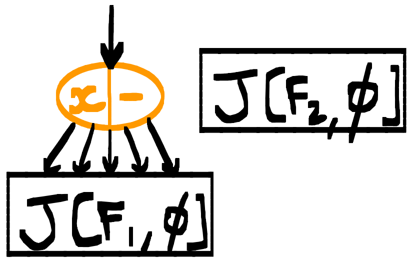


GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

- If $F = \emptyset$ and $G = \emptyset$, then $J[F,G] \leftarrow \varepsilon$

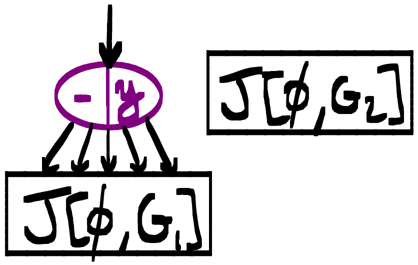
- If $F =$ 

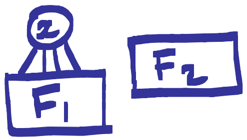
and $G = \emptyset$, then

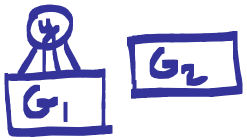
$J[F,G] \leftarrow$ 

- If $F = \emptyset$ and $G =$ 

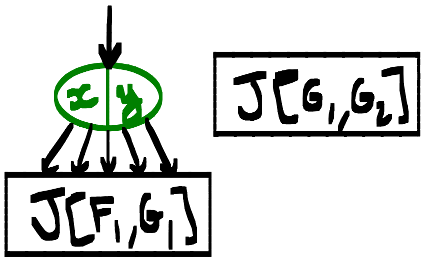
, then

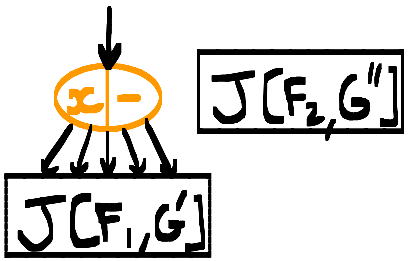
$J[F,G] \leftarrow$ 

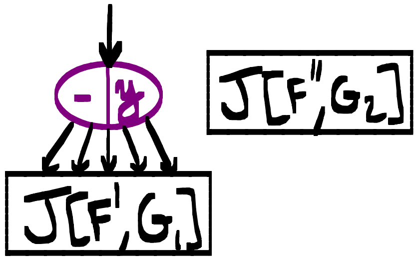
- If $F =$ 

and $G =$ 

, then

$J[F,G] \leftarrow$ 

$\mid \bigoplus_{G = G' \circ G''}$ 

$\mid \bigoplus_{F = F' \circ F''}$ 

SAMPLING

Theorem Let S and T be two trees of size n_1 and n_2.
Sampling alignments between S and T under the Gibbs-Boltzmann distribution can be done with worst-case time and space complexities $O(n_1 n_2 (n_1 + n_2)^2)$ and with average-case time and space complexities $O(n_1 n_2)$.

SAMPLING

Theorem Let S and T be two trees of size n_1 and n_2. Sampling alignments between S and T under the Gibbs-Boltzmann distribution can be done with worst-case time and space complexities $O(n_1 n_2 (n_1 + n_2)^2)$ and with average-case time and space complexities $O(n_1 n_2)$.

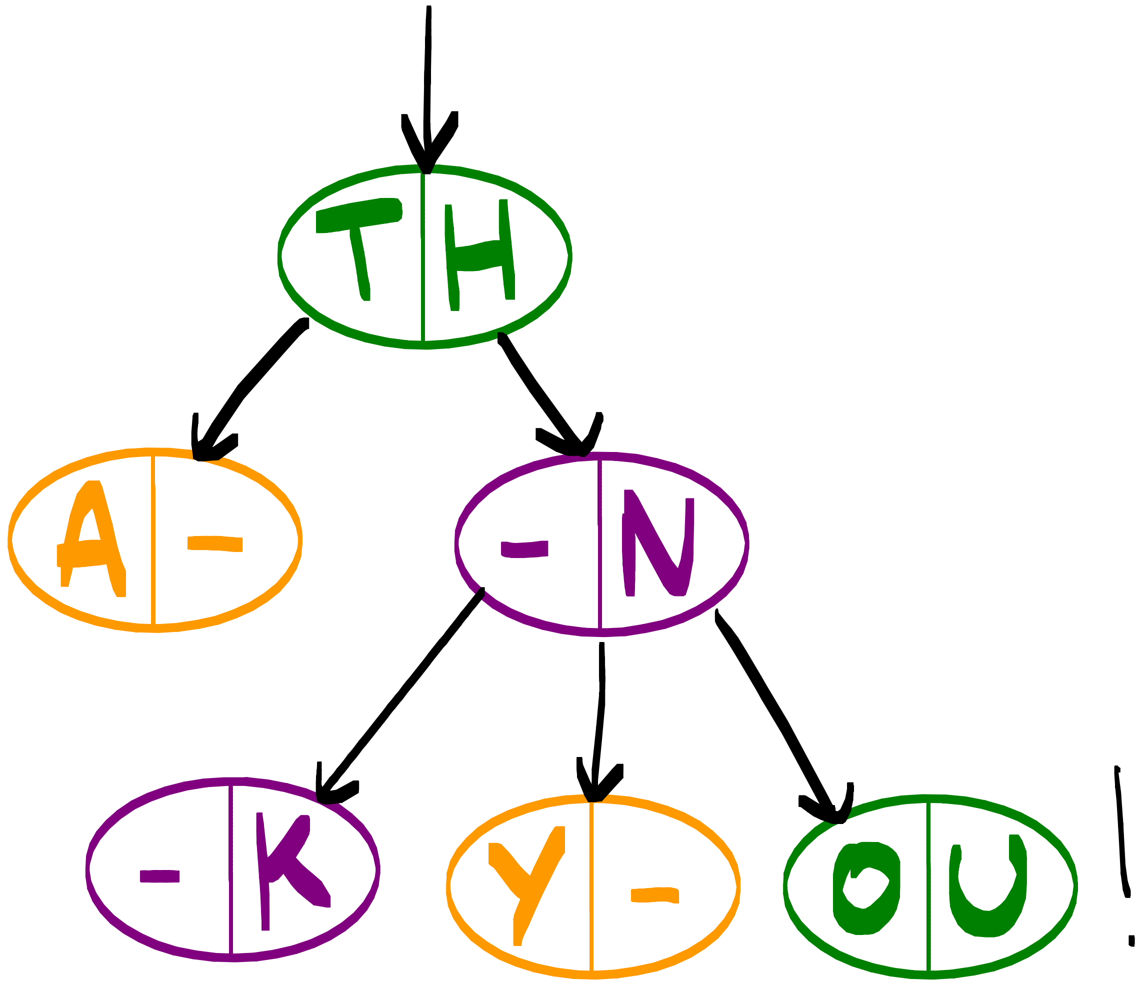
Proof inspired by

[Herrbach, Denise, Dulucq]

CONCLUSION

→ We are using our grammar and adapted dynamic programming algorithms to revisit the 3D alignments of RNA structures.

→ more general method?
new way to design dynamic programming algorithms?



T H
A -
- N
- K
Y -
O U
!