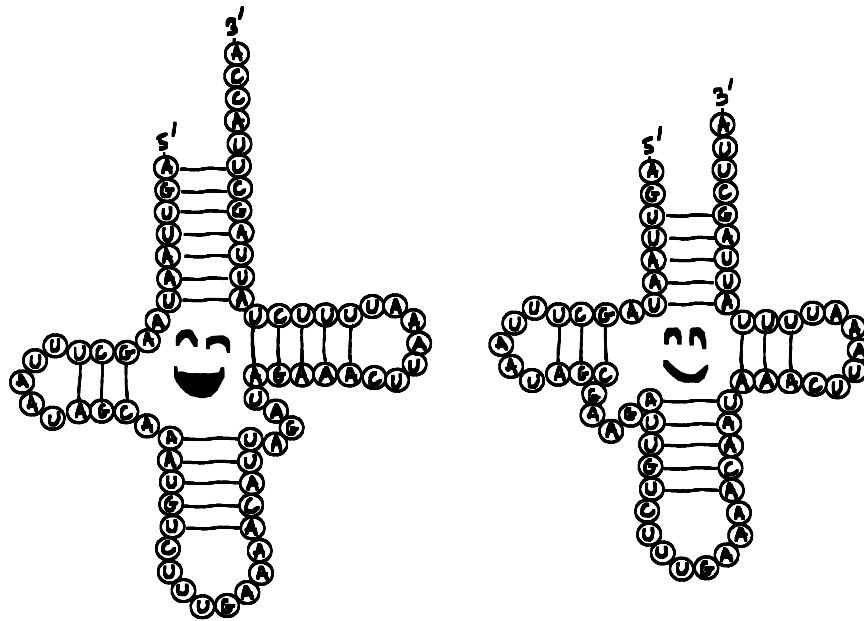

COUNTING, GENERATING AND SAMPLING TREE ALIGNMENTS

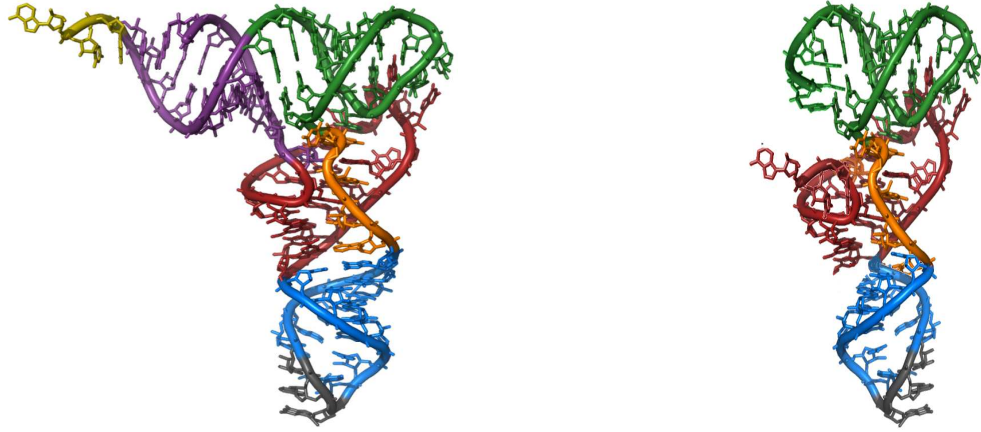
Cedric CHAUVÉ (Simon Fraser University, Vancouver)
Julien COURTIEL (PIMS/Univ. of British Columbia, Vancouver)
Yann PONTY (CNRS/LIX)



LRI, January 14th 2016

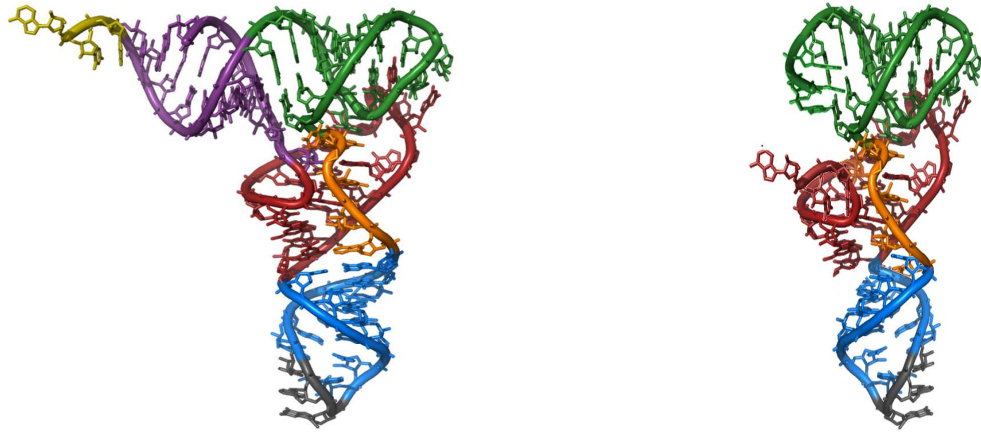
MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?



MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?



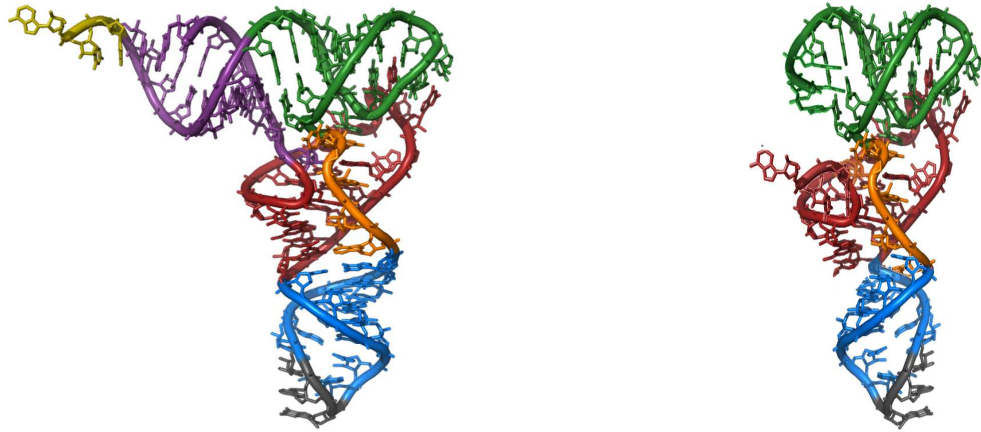
First idea: compare nucleic acid sequences.

RNA 1: AUUCGAUUA...

RNA 2: ACCAUGAUUA...

MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?



First idea: compare nucleic acid sequences.
→ sequence alignment

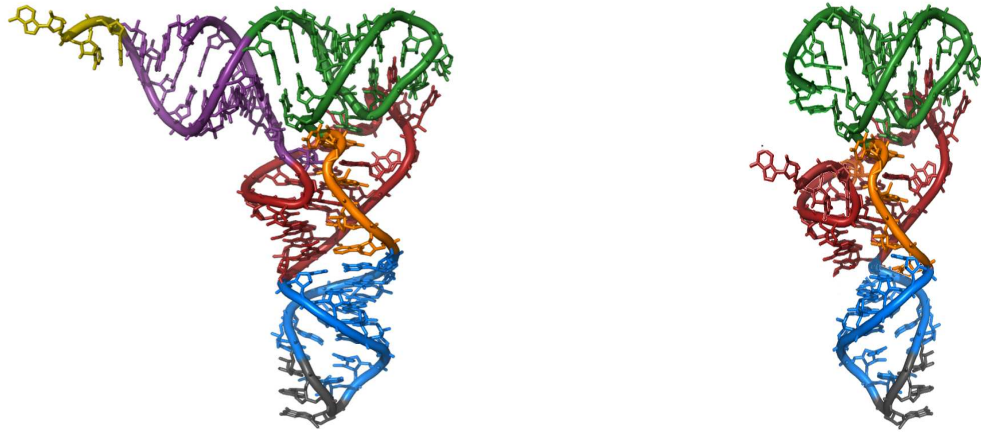
RNA 1: AUUCGAUUA...

RNA 2: ACCAUGAUUA...

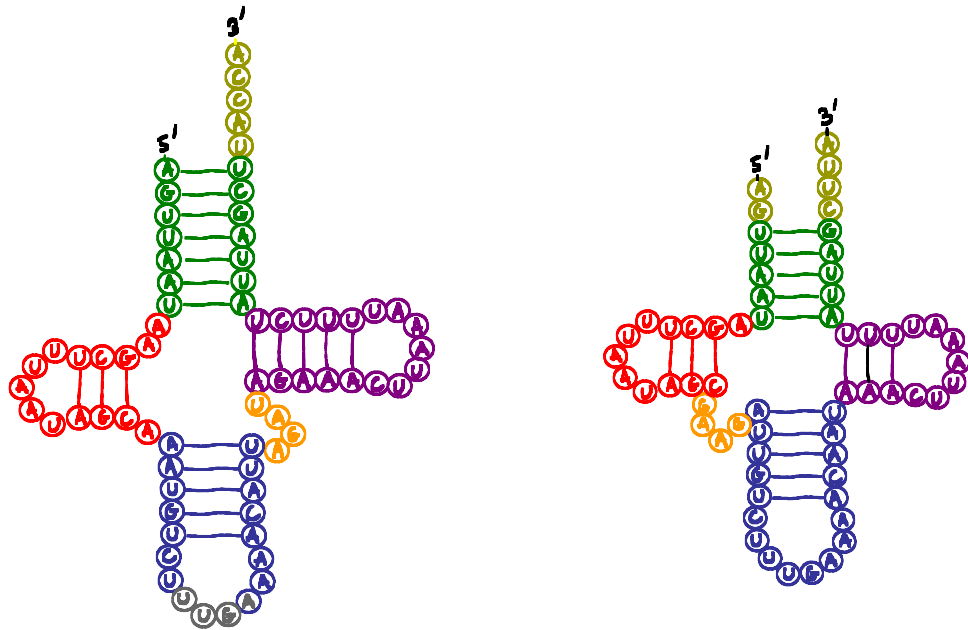
alignment: $\begin{pmatrix} A \\ A \end{pmatrix} \begin{pmatrix} U \\ - \end{pmatrix} \begin{pmatrix} - \\ C \end{pmatrix} \begin{pmatrix} U \\ - \end{pmatrix} \begin{pmatrix} C \\ C \end{pmatrix} \begin{pmatrix} - \\ A \end{pmatrix} \begin{pmatrix} - \\ U \end{pmatrix} \begin{pmatrix} G \\ G \end{pmatrix}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

MOTIVATION: RNA COMPARISON

Question: how to measure similarity between two RNAs?

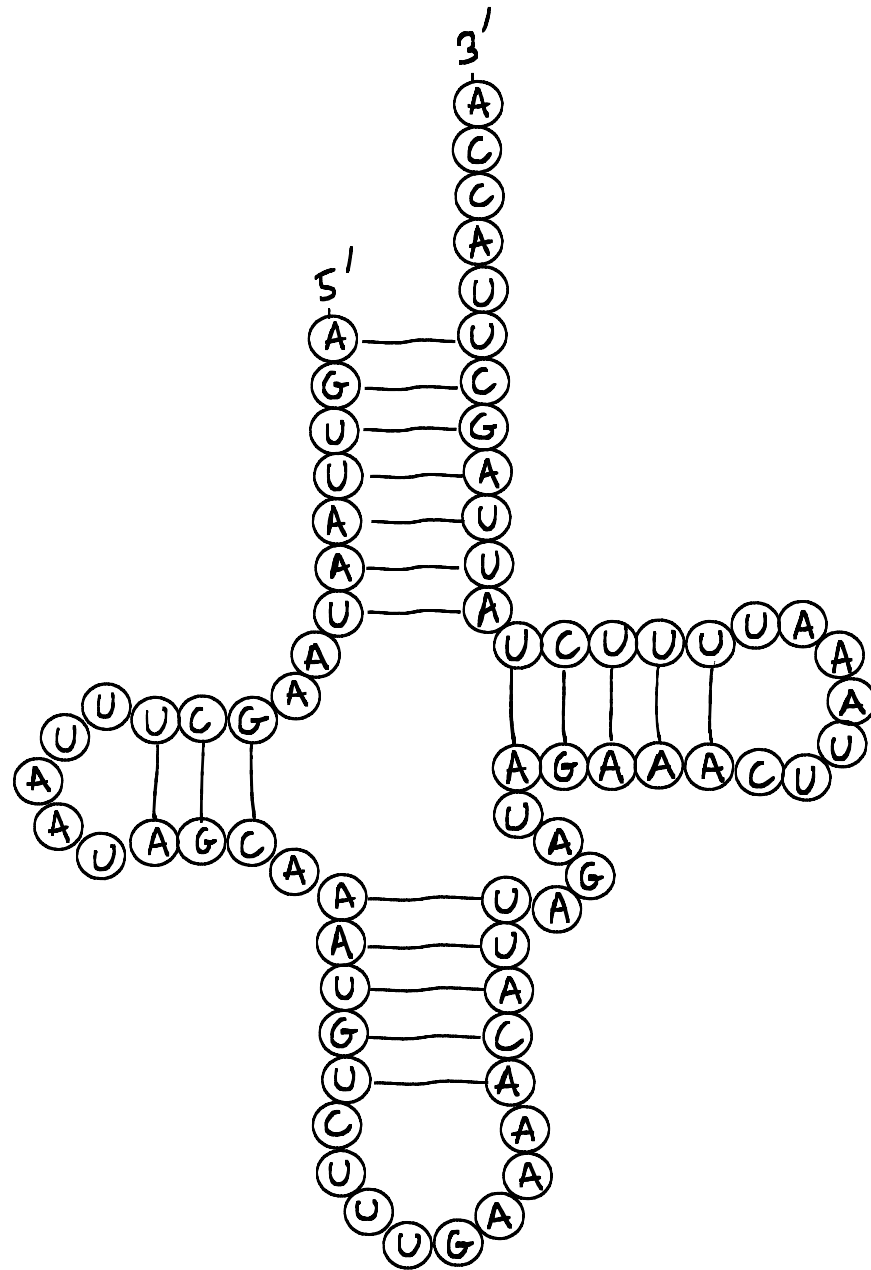


Second idea: compare secondary structures.

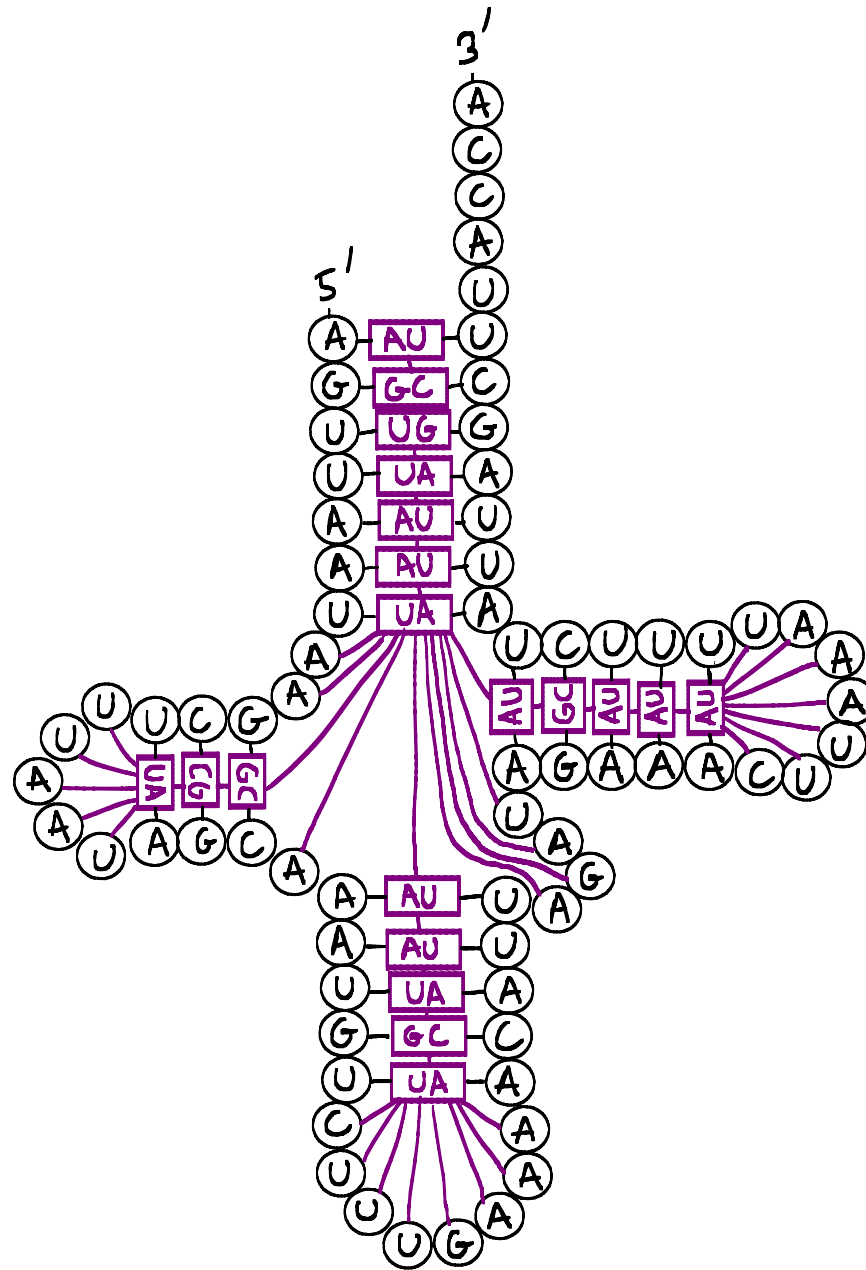


→ notion of
tree alignment
[Jiang, Wang,
Zhang]

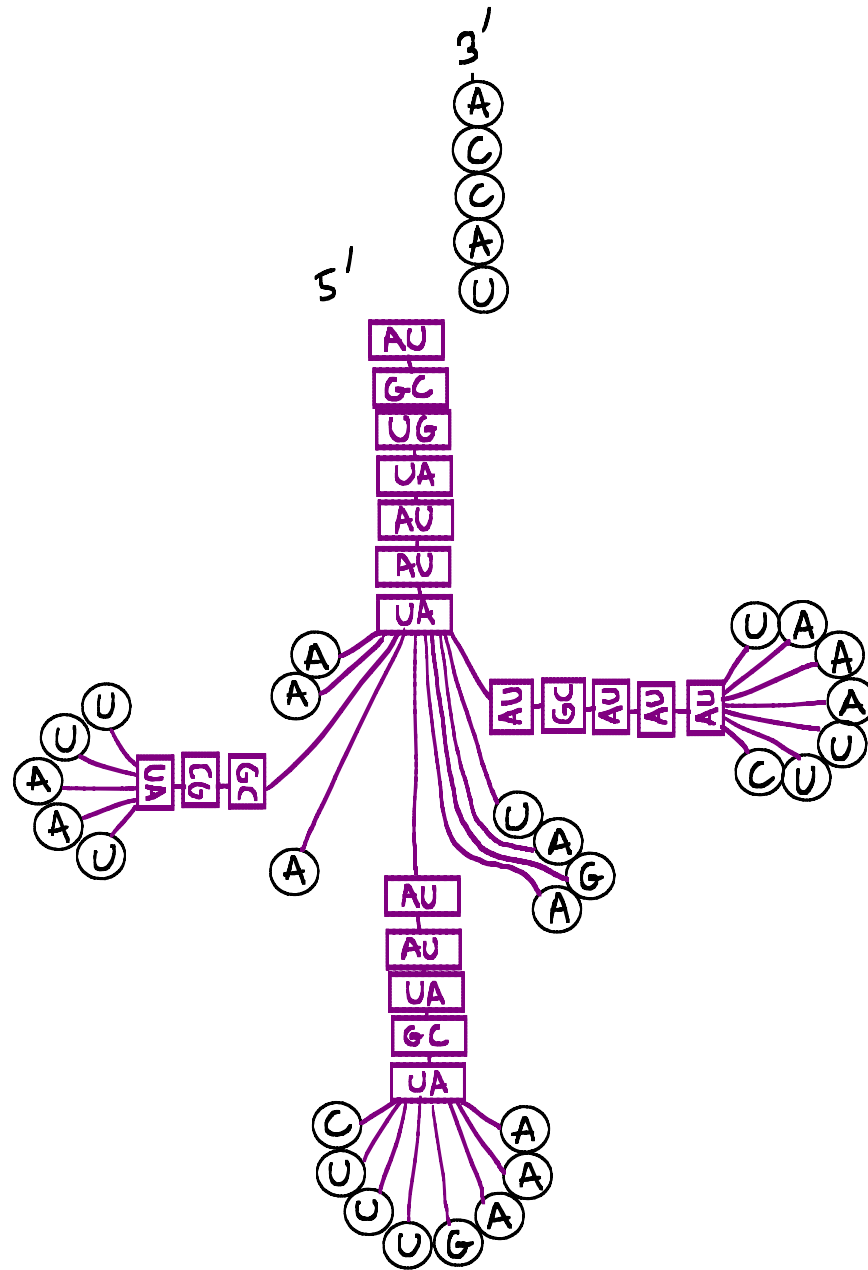
FROM SECONDARY STRUCTURES TO TREES



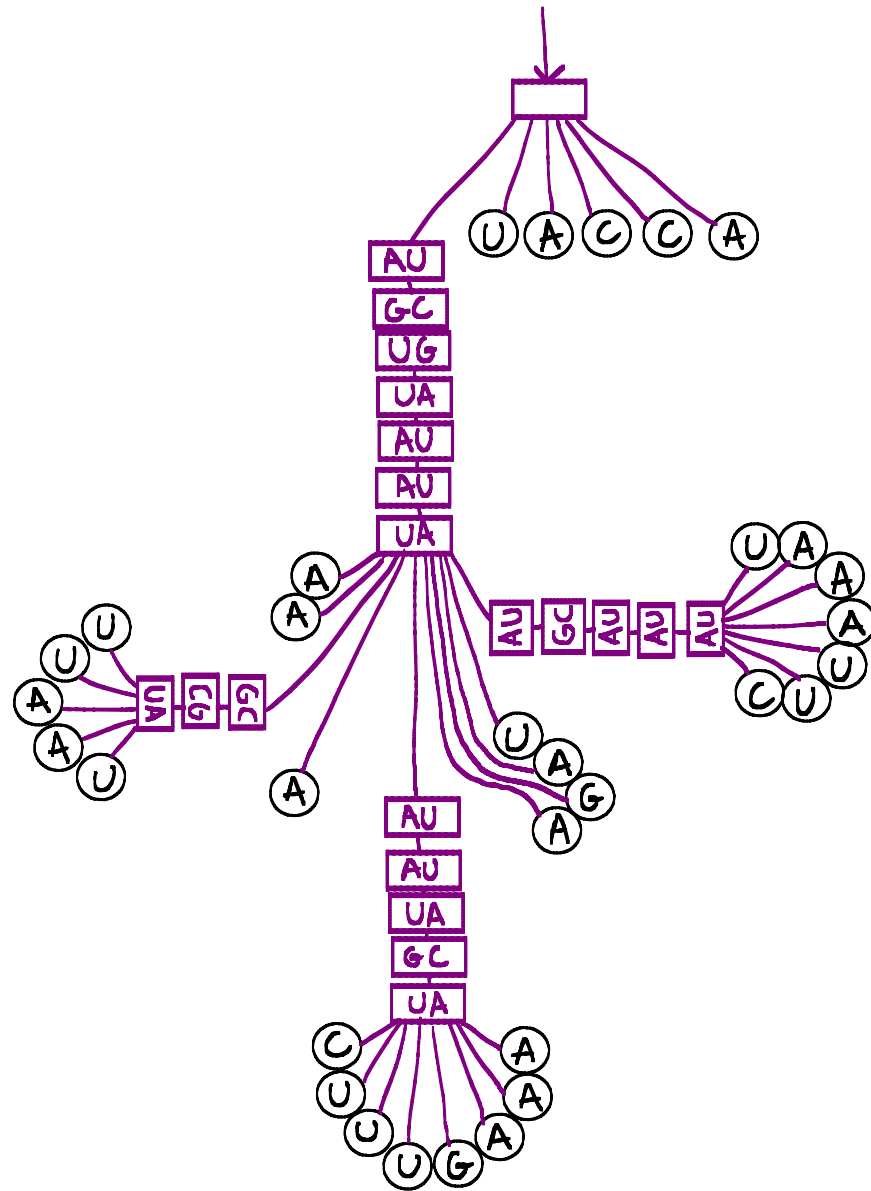
FROM SECONDARY STRUCTURES TO TREES



FROM SECONDARY STRUCTURES TO TREES

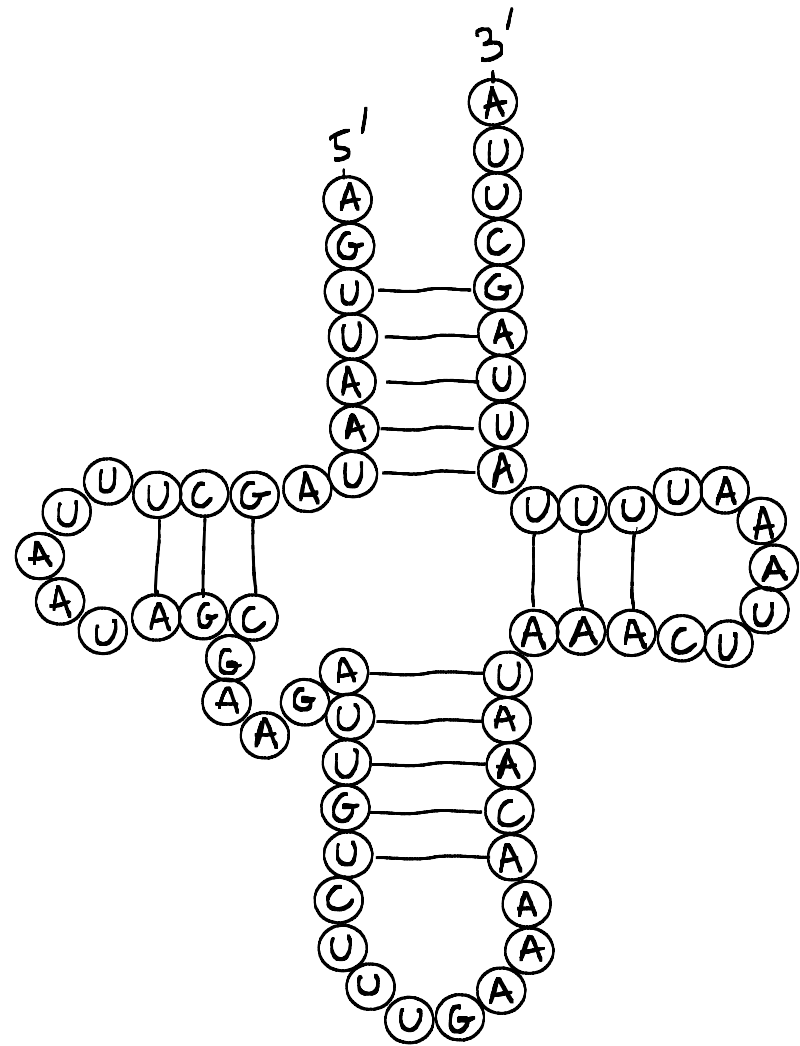
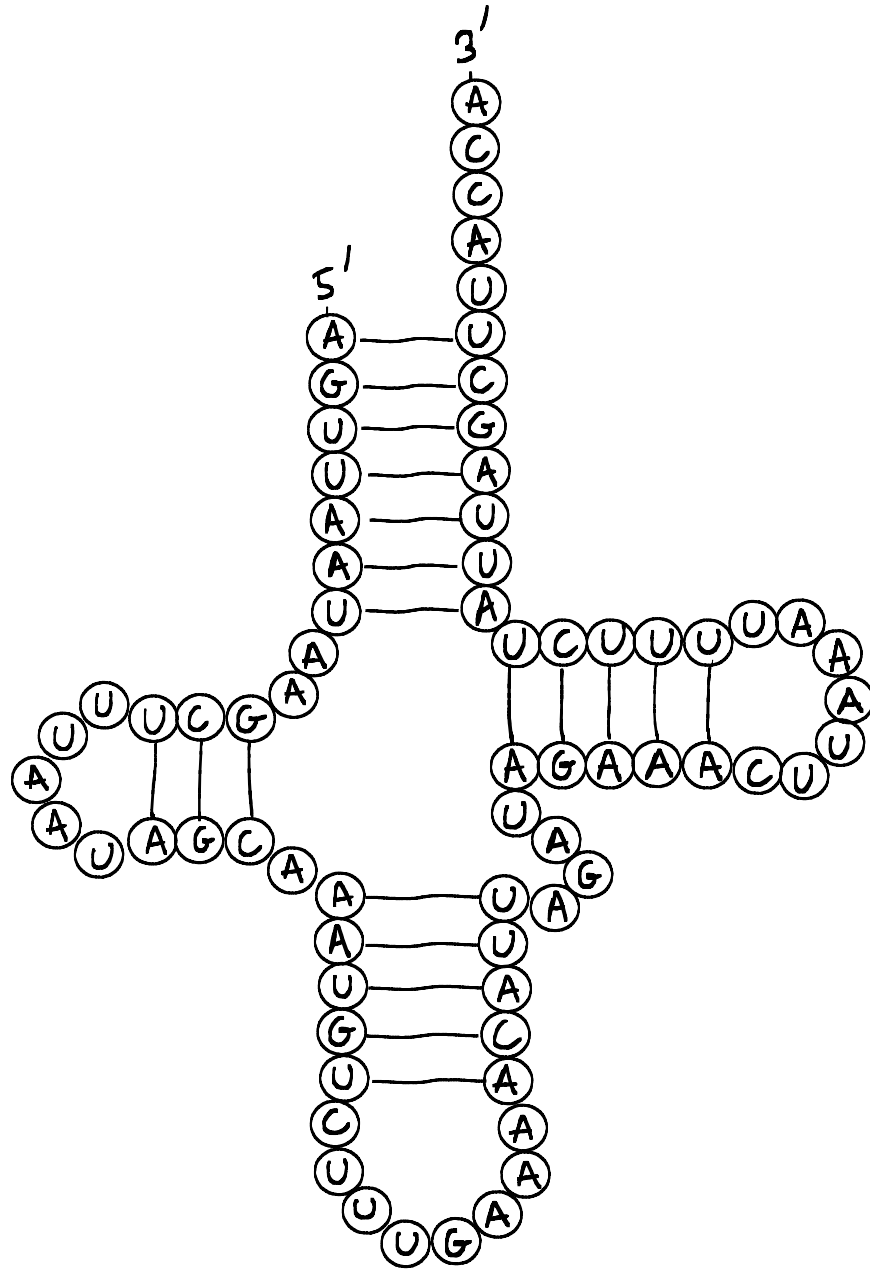


FROM SECONDARY STRUCTURES TO TREES



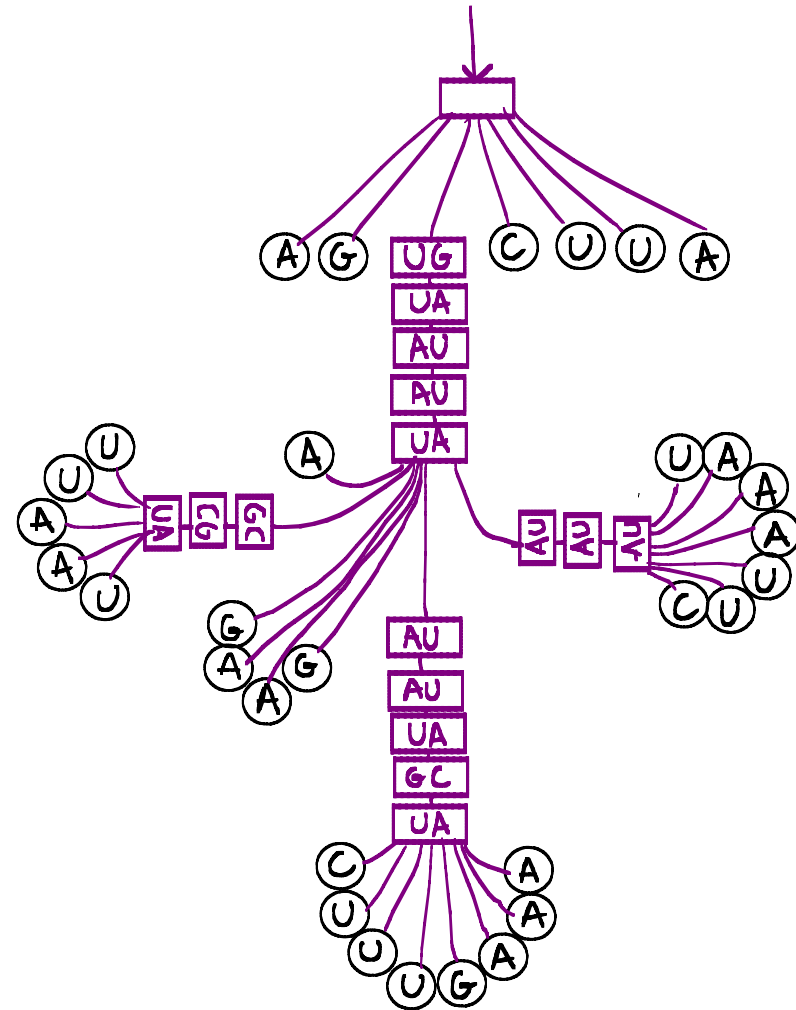
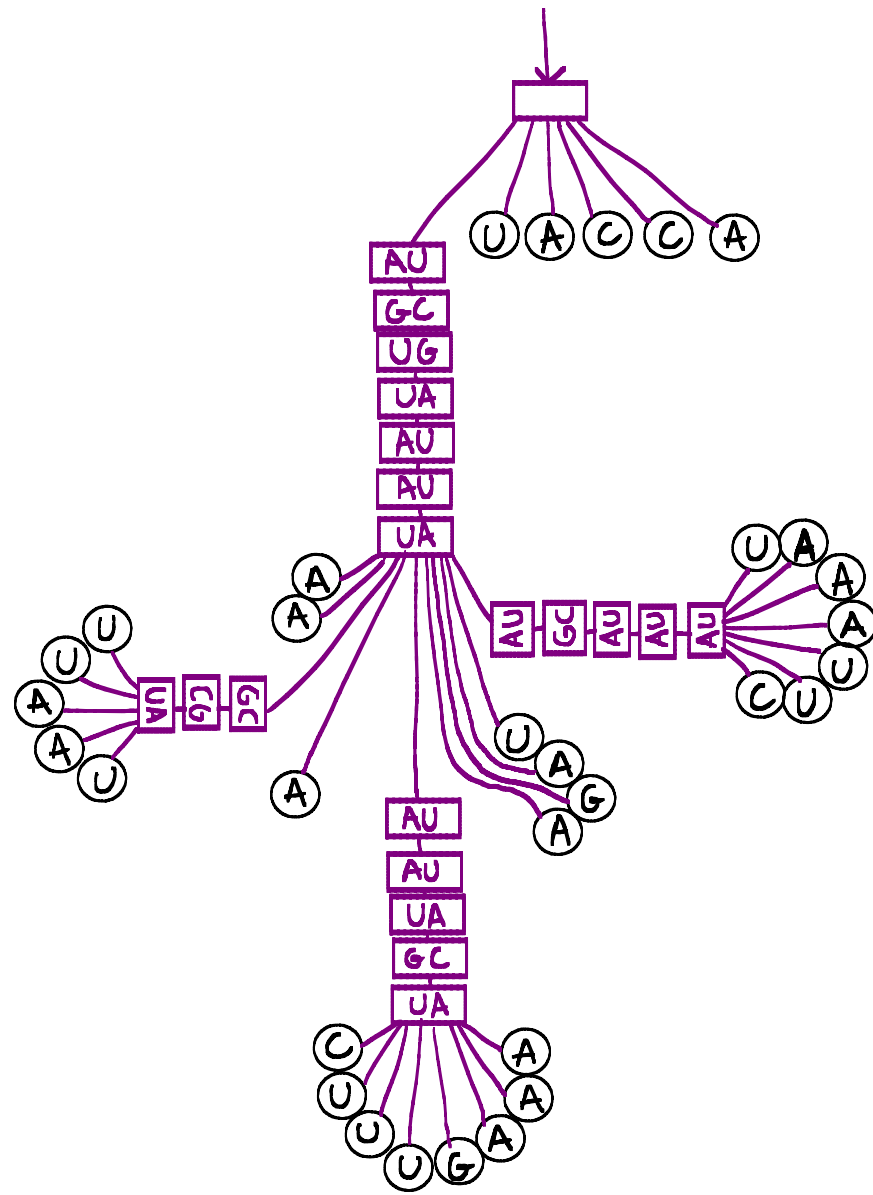
FROM SECONDARY STRUCTURES TO TREES

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



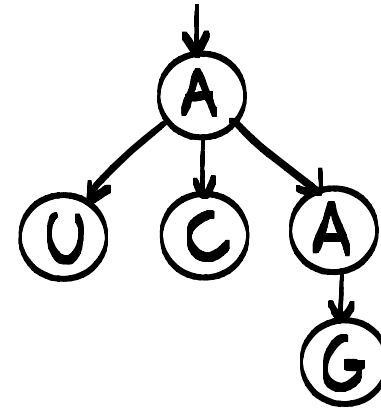
FROM SECONDARY STRUCTURES TO TREES

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



TREES AND SUPERTREES

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

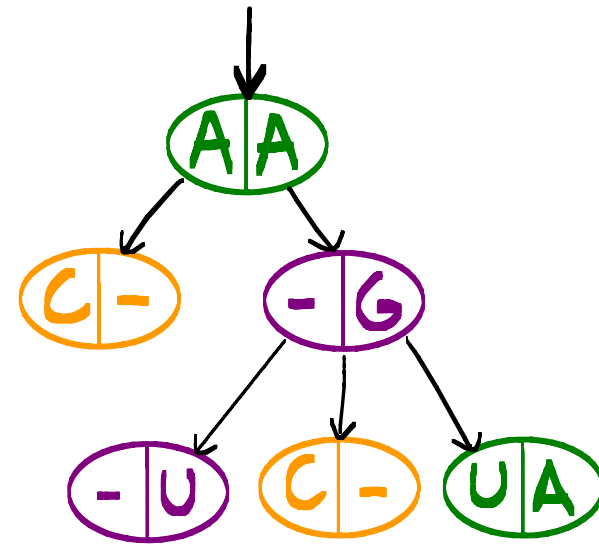


Supertree = tree with 3 types of vertices:

$(X|Y)$ (mis)match

$(X|-)$ insertion

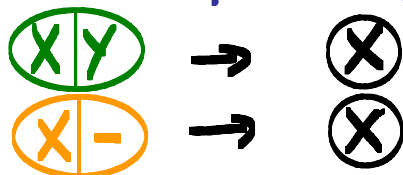
$(-|Y)$ deletion



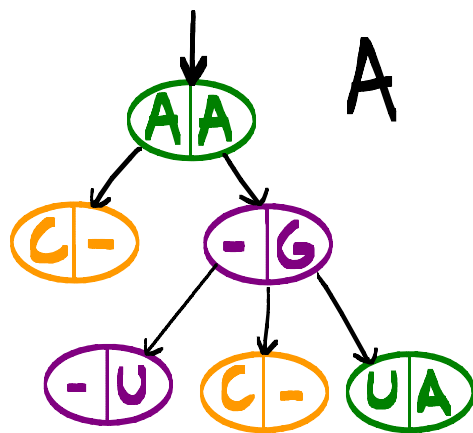
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

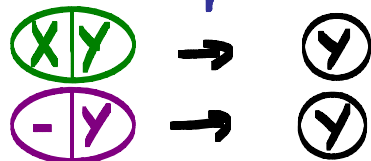
$\Pi_1(A)$ = tree
obtained by changing



and removing -|Y .



$\Pi_2(A)$ = tree
obtained by changing

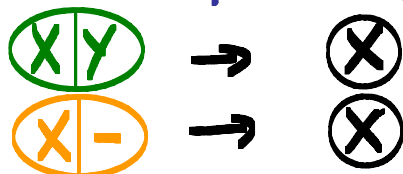


and removing X|- .

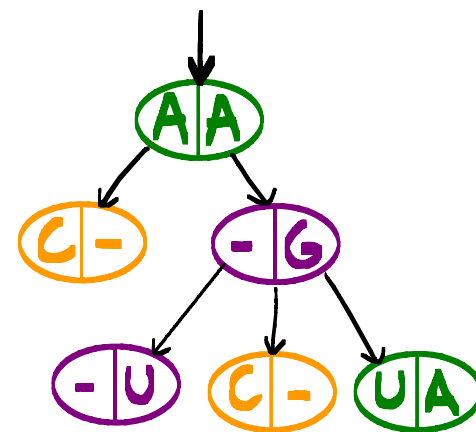
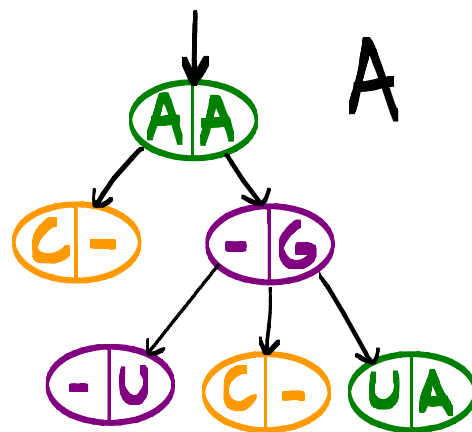
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

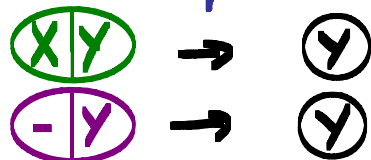
$\Pi_1(A)$ = tree
obtained by changing



and removing -|Y .



$\Pi_2(A)$ = tree
obtained by changing

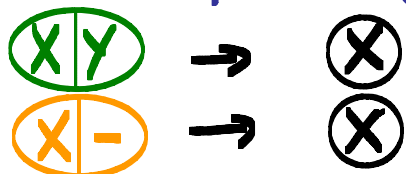


and removing X|- .

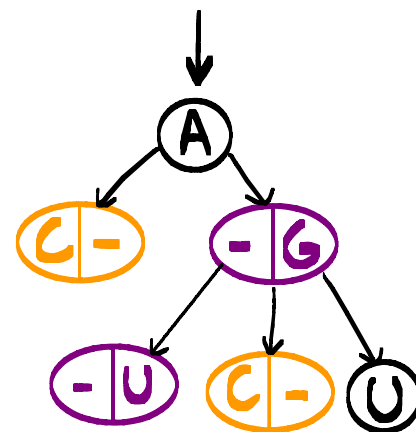
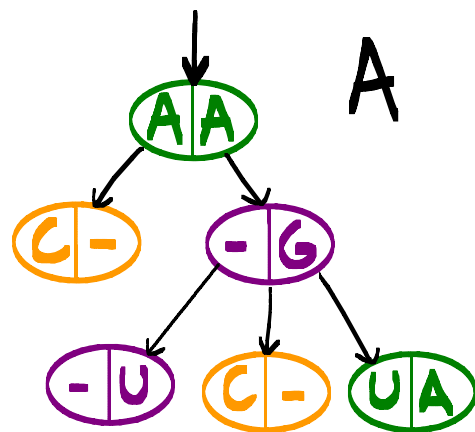
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

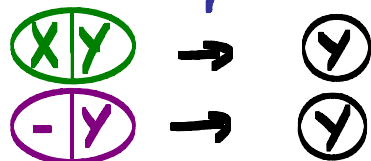
$\Pi_1(A)$ = tree
obtained by changing



and removing -|Y .



$\Pi_2(A)$ = tree
obtained by changing

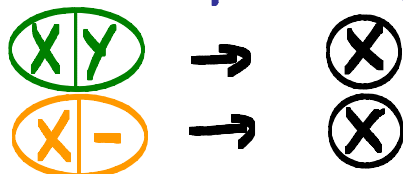


and removing X|- .

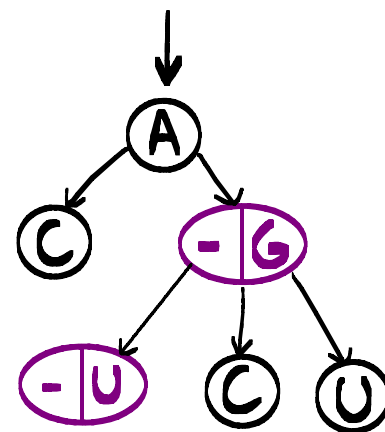
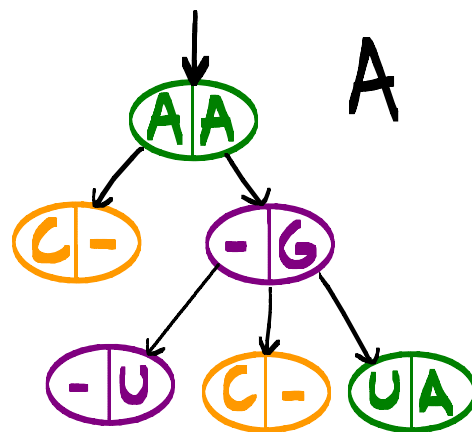
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

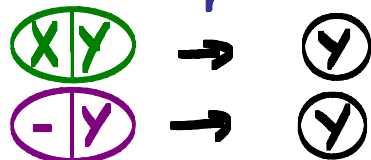
$\Pi_1(A)$ = tree
obtained by changing



and removing -|Y .



$\Pi_2(A)$ = tree
obtained by changing

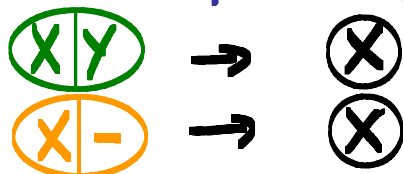


and removing X|- .

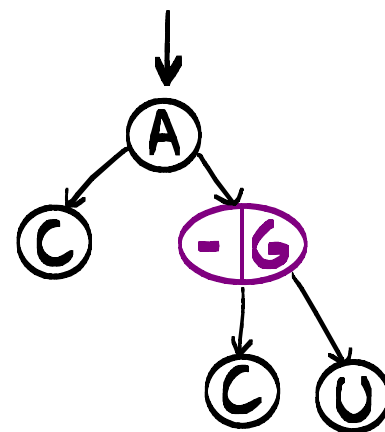
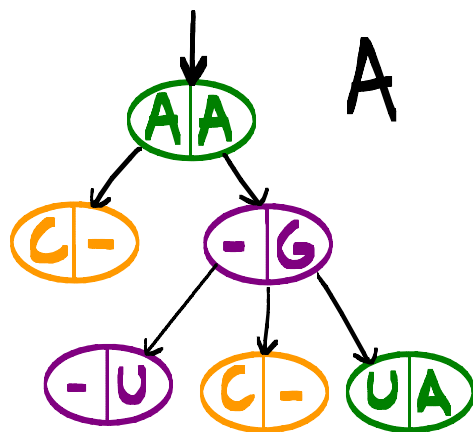
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

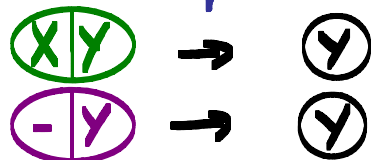
$\Pi_1(A)$ = tree
obtained by changing



and removing -|Y .



$\Pi_2(A)$ = tree
obtained by changing

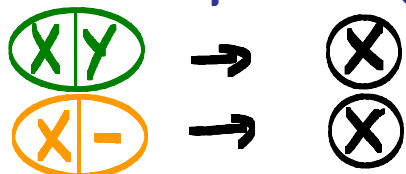


and removing X|- .

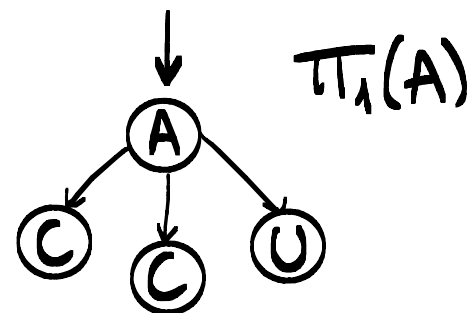
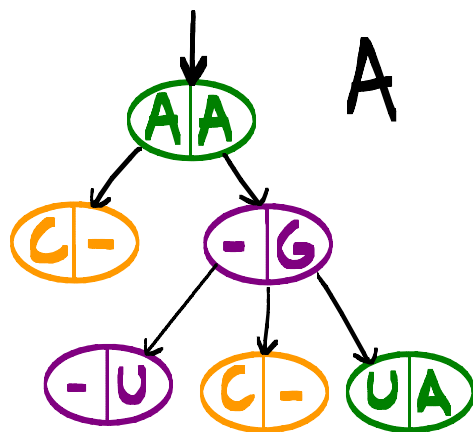
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

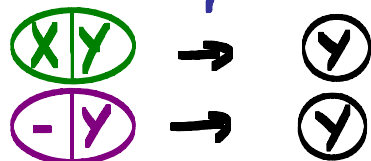
$\pi_1(A)$ = tree
obtained by changing



and removing $\textcircled{-|Y}$.



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

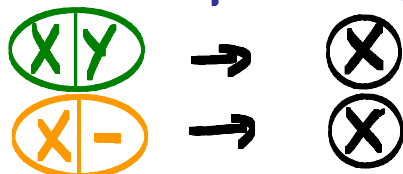


and removing $\textcircled{X|-}$.

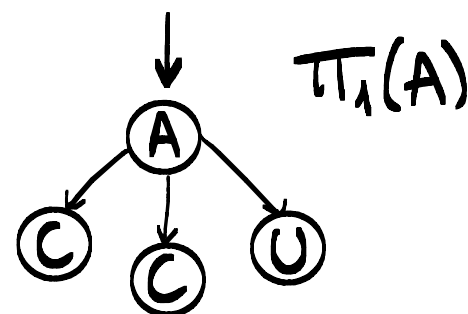
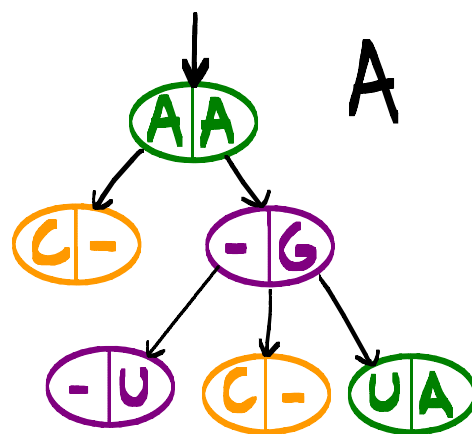
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

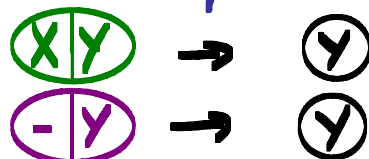
$\pi_1(A)$ = tree
obtained by changing



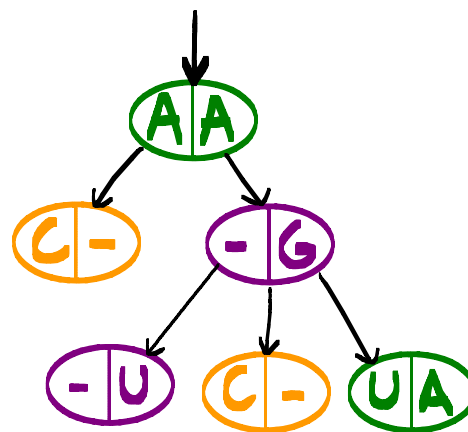
and removing $\begin{array}{|c|c|} \hline - & Y \\ \hline \end{array}$.



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



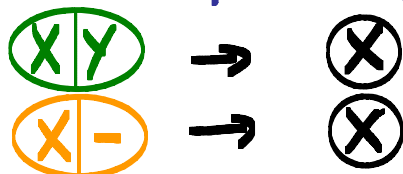
and removing $\begin{array}{|c|c|} \hline X & - \\ \hline \end{array}$.



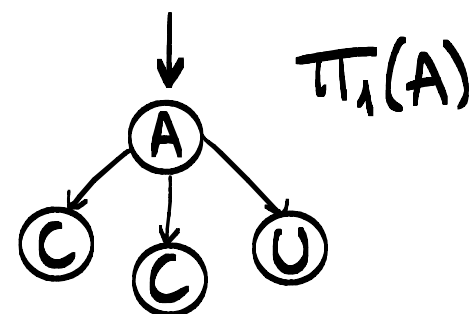
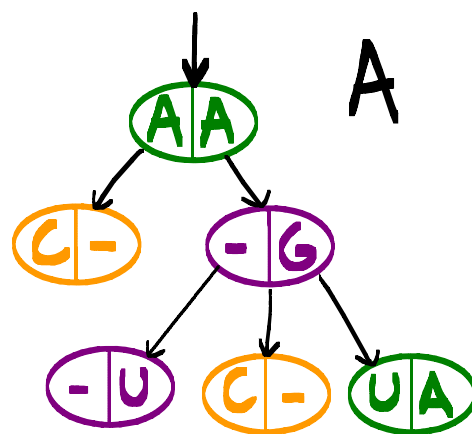
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

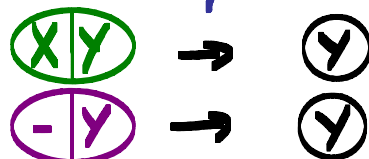
$\pi_1(A)$ = tree
obtained by changing



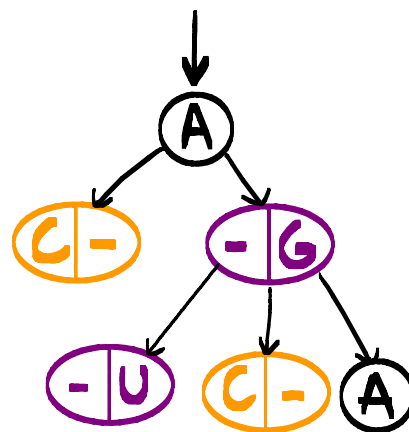
and removing $\textcircled{-|Y}$.



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



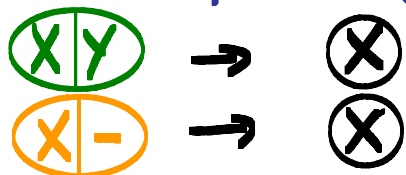
and removing $\textcircled{X|-}$.



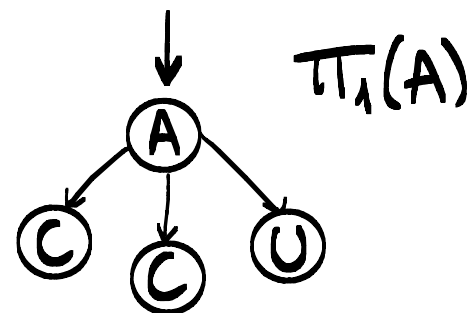
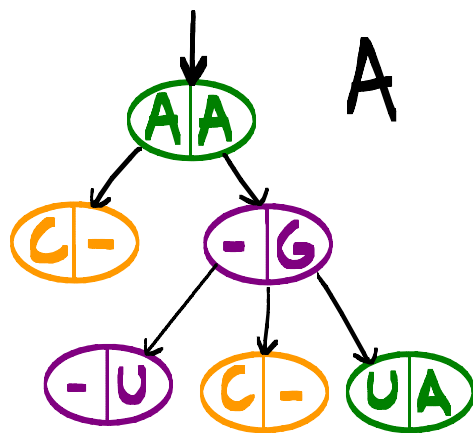
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

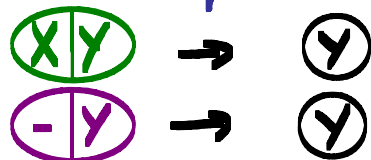
$\pi_1(A)$ = tree
obtained by changing



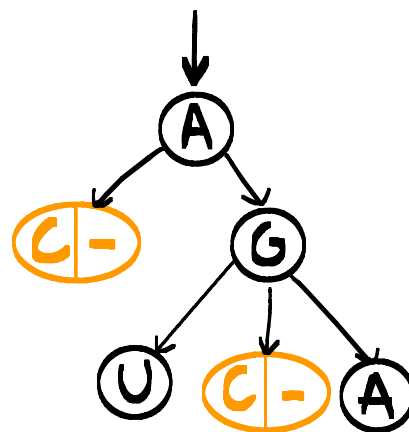
and removing -Y .



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



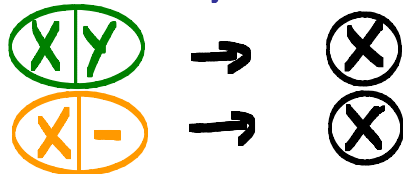
and removing X- .



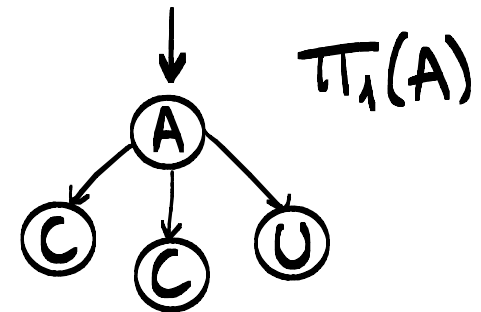
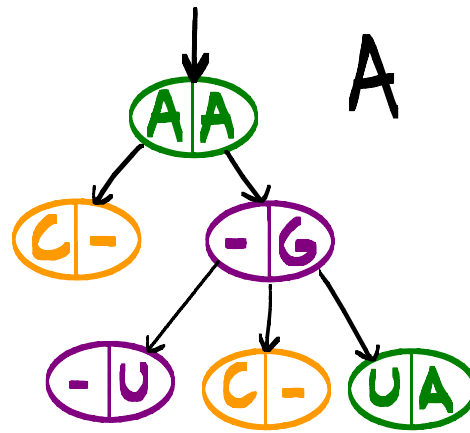
SUPERTREES INDUCE TREE ALIGNMENTS

Let A be a supertree,

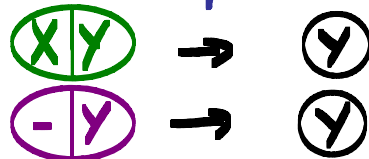
$\pi_1(A)$ = tree
obtained by changing



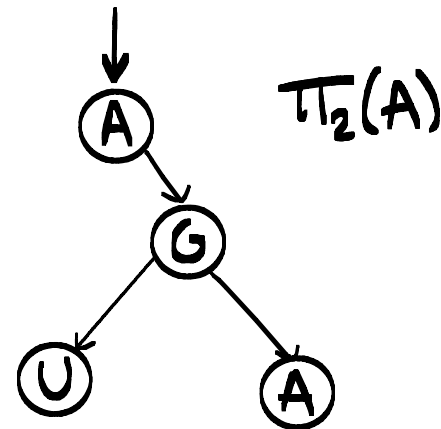
and removing $\textcircled{-|Y}$.



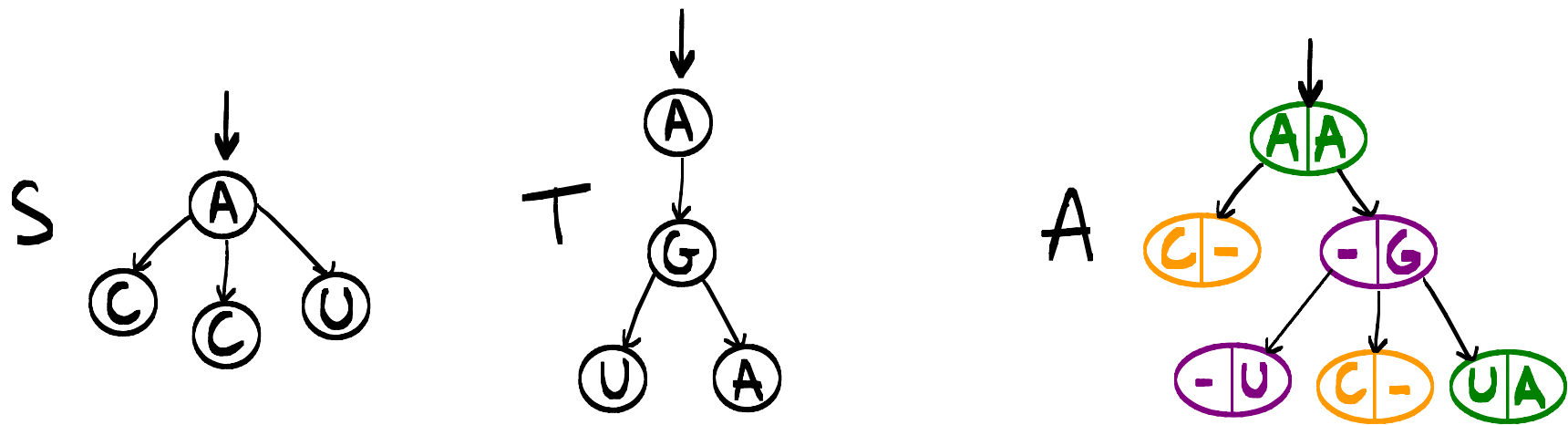
$\pi_2(A)$ = tree
obtained by changing



and removing $\textcircled{X|-}$.

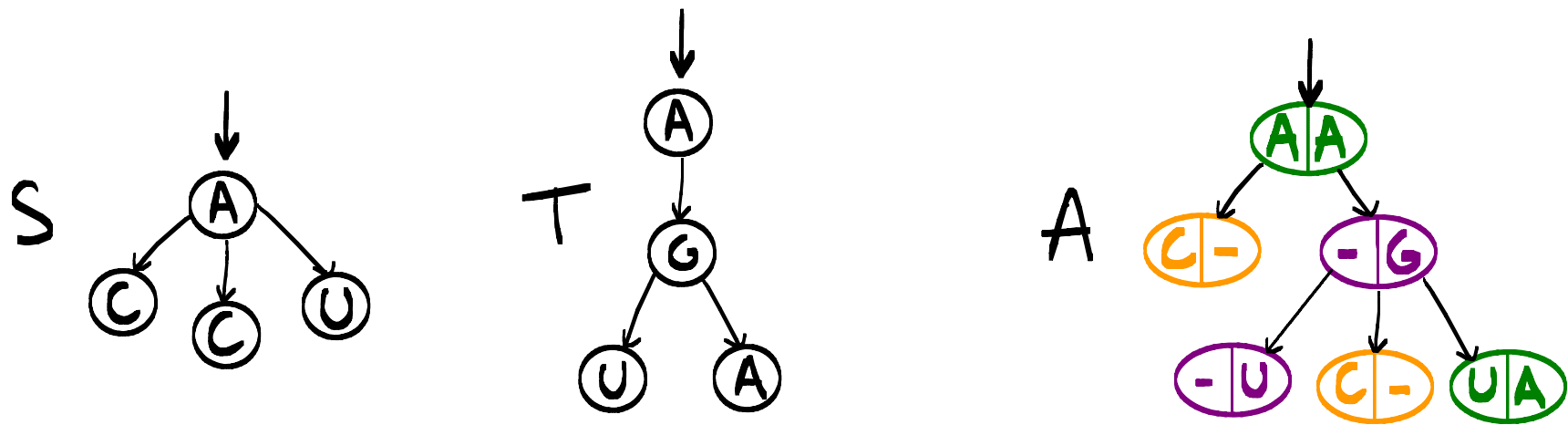


SUPERTREES INDUCE TREE ALIGNMENTS



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 $\pi_2(A) = T$.

$\text{cost}(A) = \text{nb of insertions} + \text{deletions} + \text{mismatches}$
(can be changed more complicated models)

CONNECTION WITH SEQUENCE ALIGNMENTS

Tree alignments generalize sequence alignments.

SEQUENCES

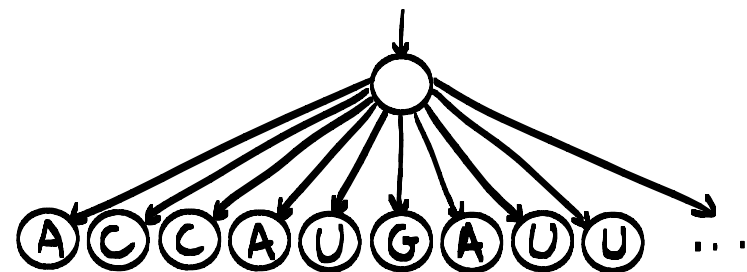
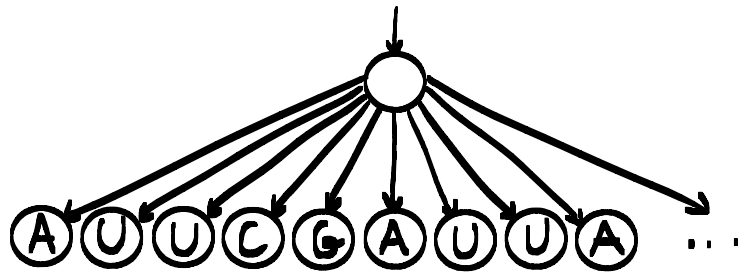
AUUCGAUUA...

ACCAUGAUUA...

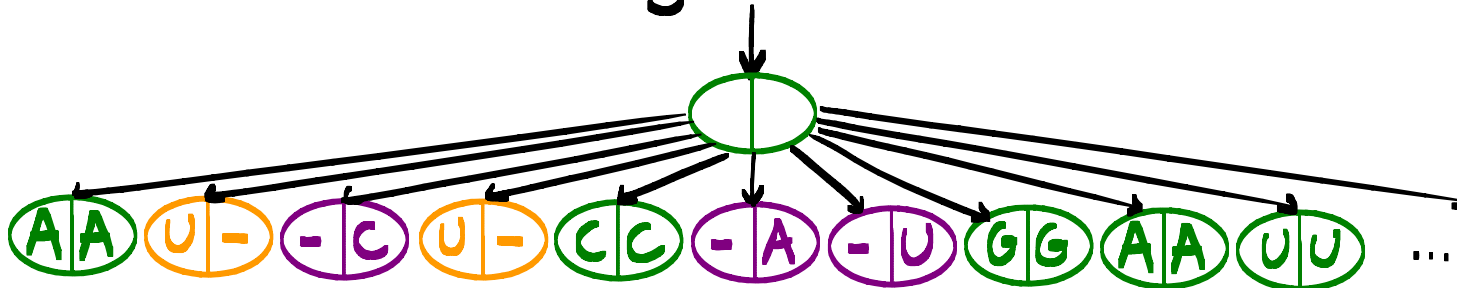
alignment:

(A)(U)(-)(U)(C)(-)(-)(G)(A)(U)(U)(A) ...
(A)(-)(C)(-)(C)(A)(U)(G)(A)(U)(U)(A) ...

TREES

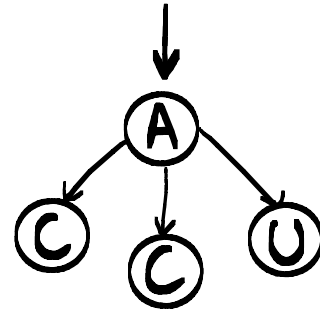


alignment:

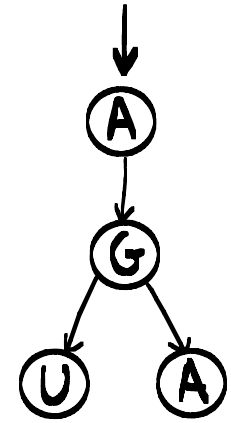


SPACE OF ALIGNMENTS

Which alignment between
is the most likely?

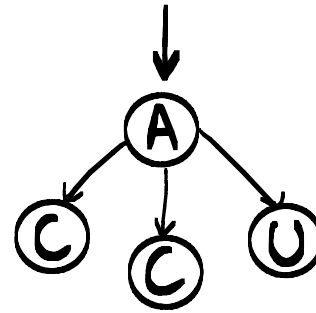


and

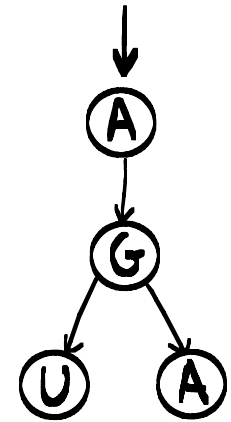


SPACE OF ALIGNMENTS

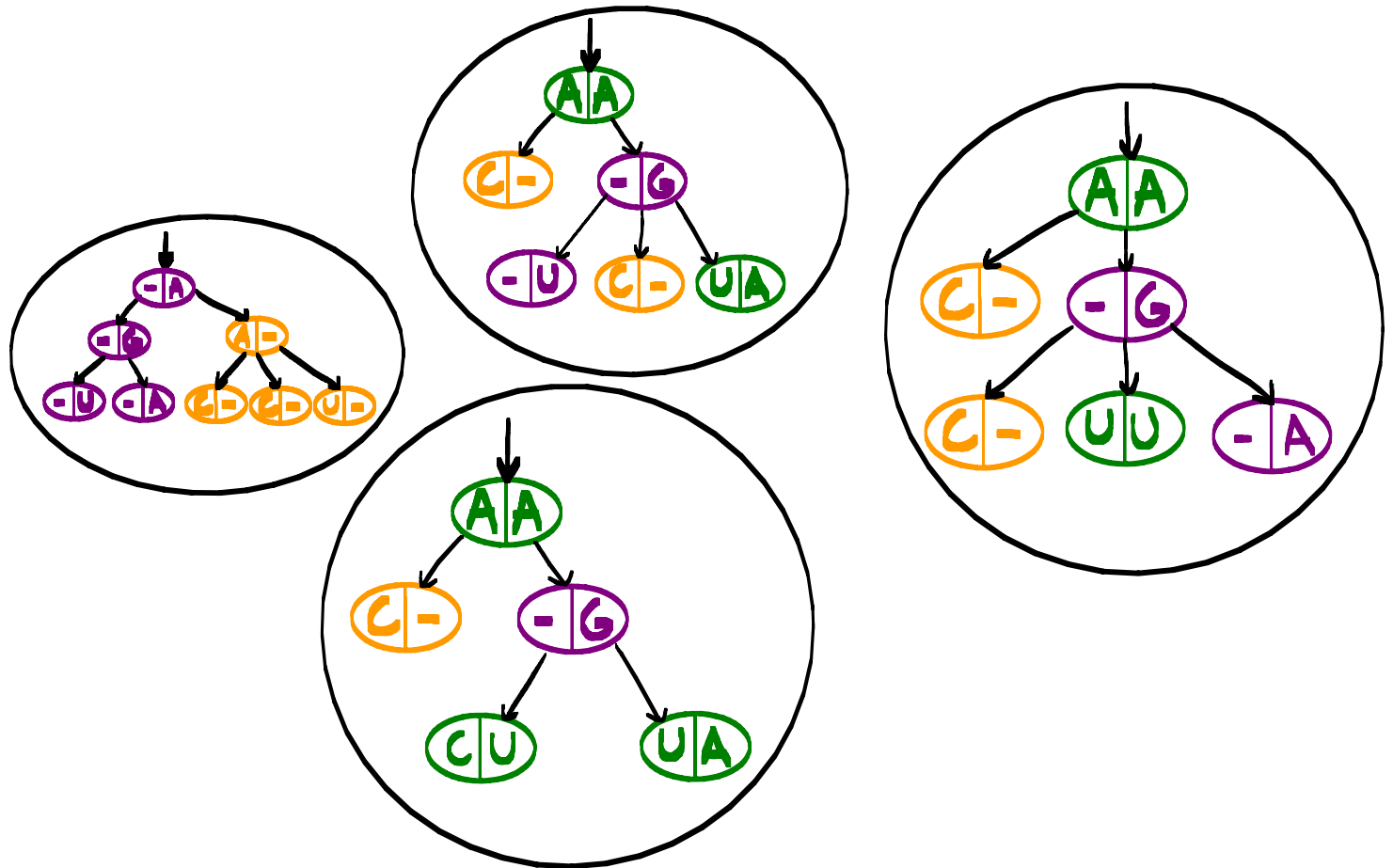
Which alignment between
is the most likely?



and

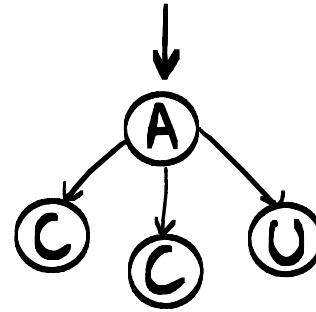


probability of
an alignment A
 $\propto e^{-\frac{\text{cost}(A)}{k}}$
(Gibbs-Boltzmann
distribution)

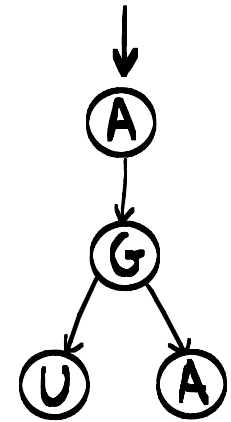


SPACE OF ALIGNMENTS

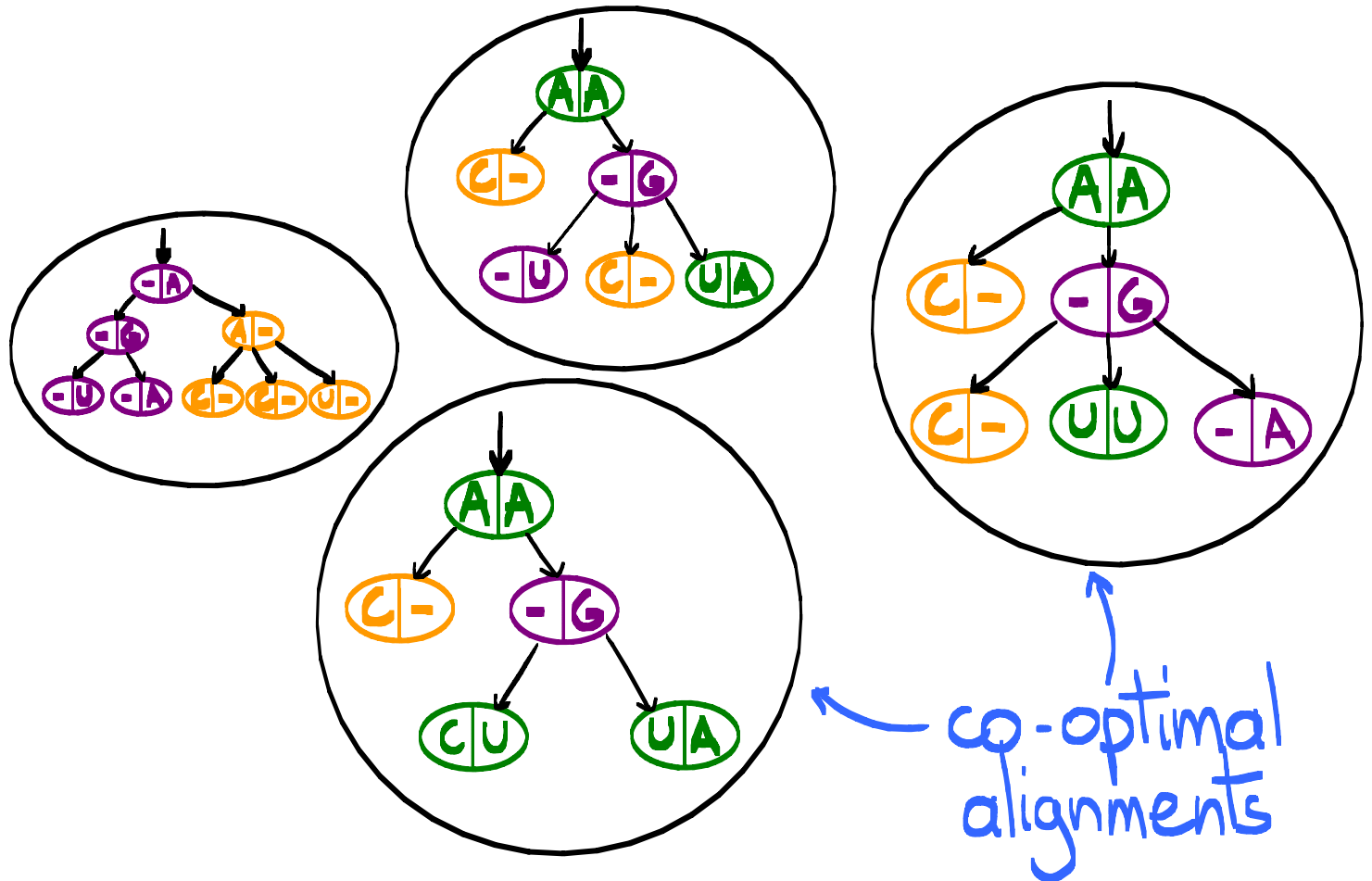
Which alignment between
is the most likely?



and



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



SPACE OF ALIGNMENTS

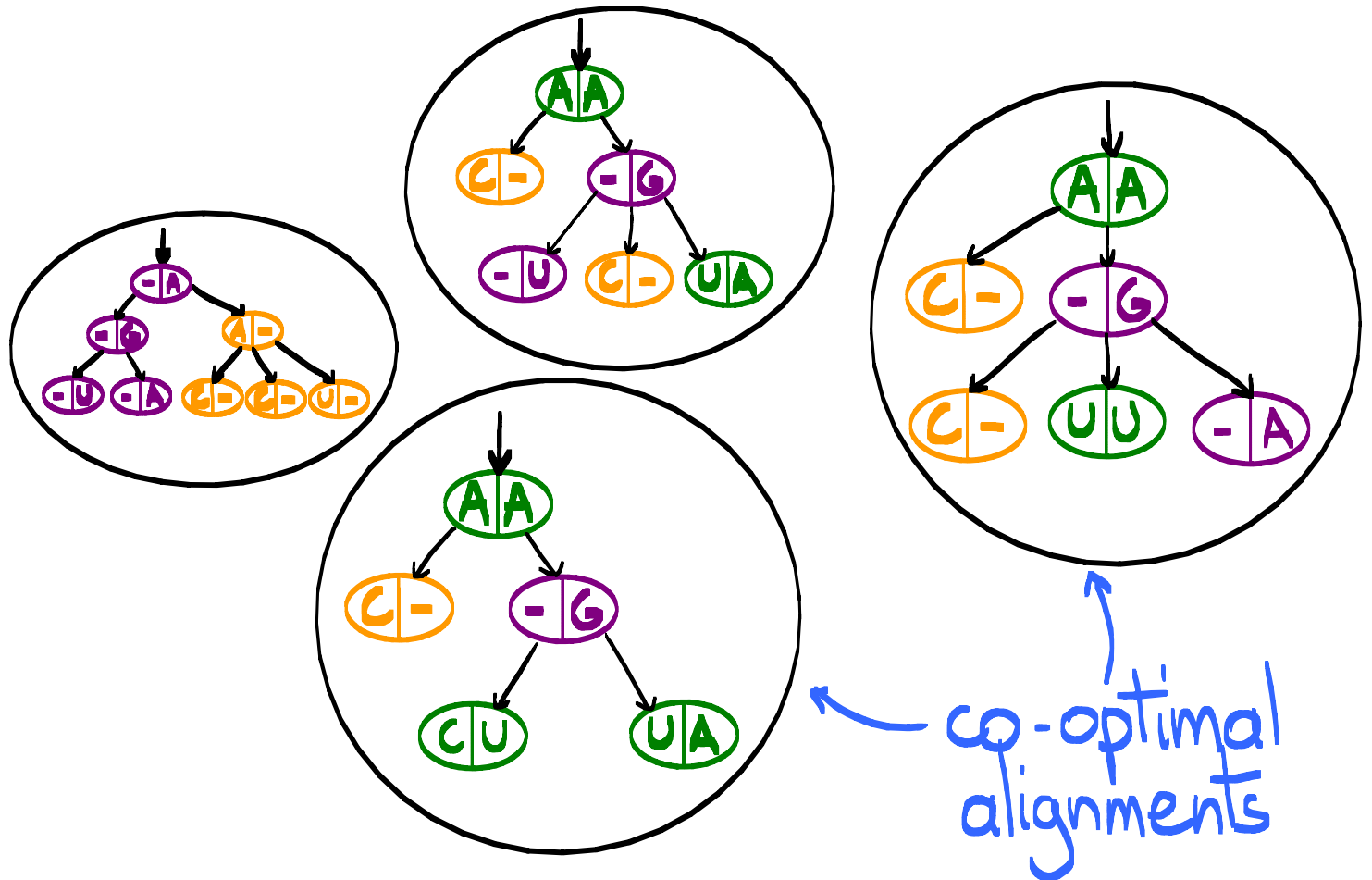
Why finding one optimal alignment may be inadequate:

- ▶ 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.

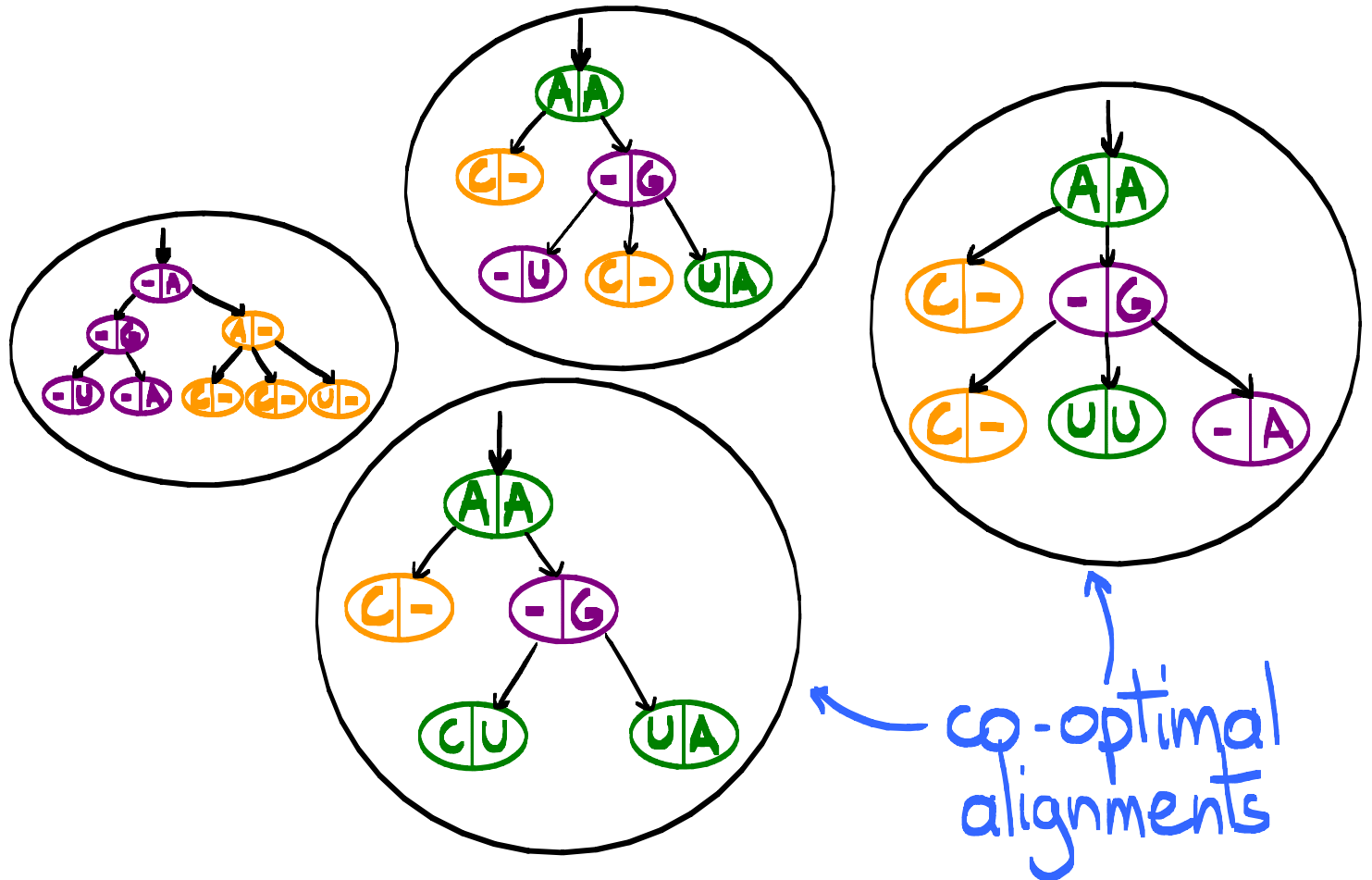
probability of an alignment A
 $\propto e^{-\frac{\text{cost}(A)}{k}}$
(Gibbs-Boltzmann distribution)



SPACE OF ALIGNMENTS

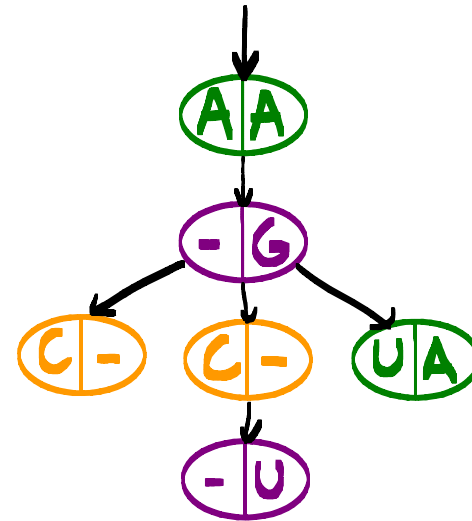
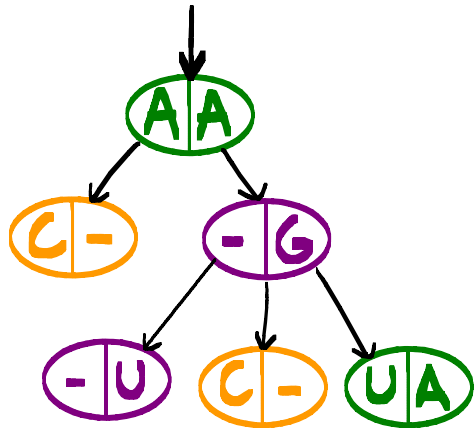
Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

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

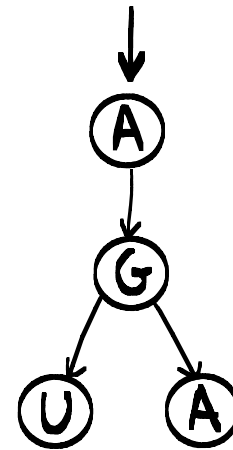
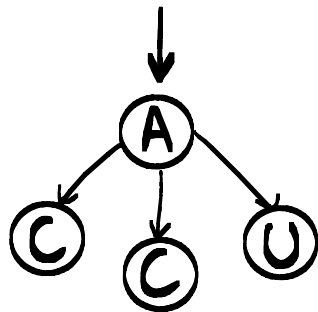


AMBIGUITY OF ALIGNMENTS

The two supertrees

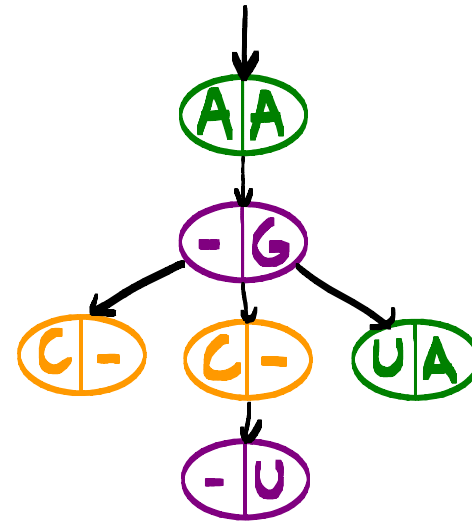
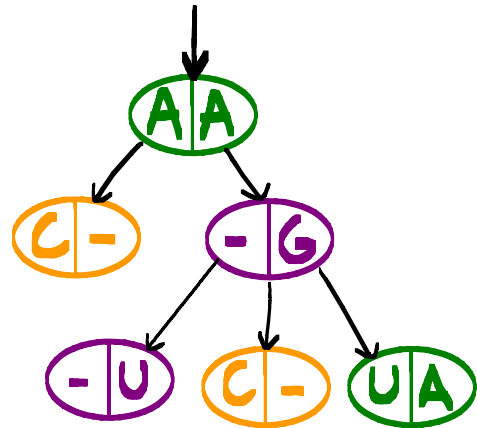


induce the same alignment between the trees

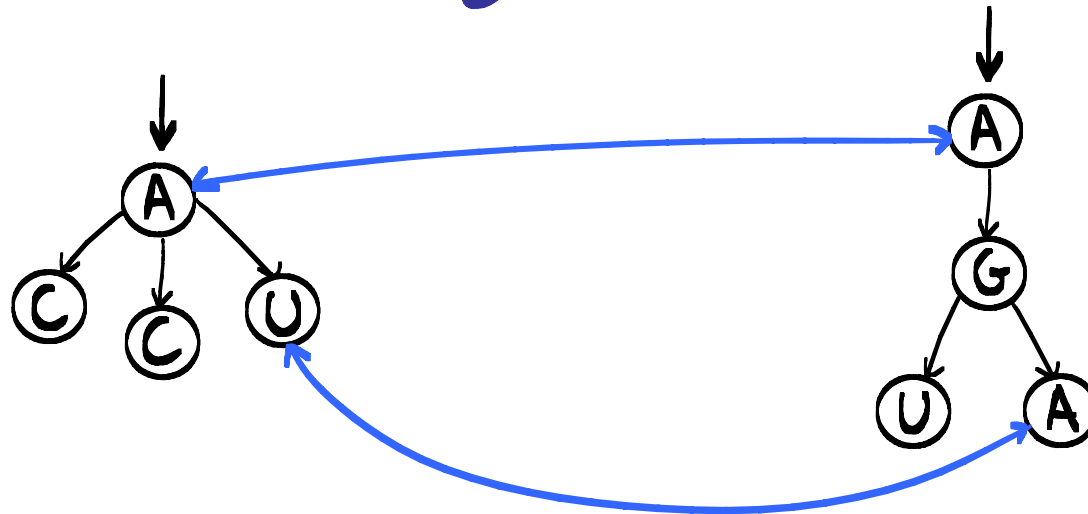


AMBIGUITY OF ALIGNMENTS

The two supertrees

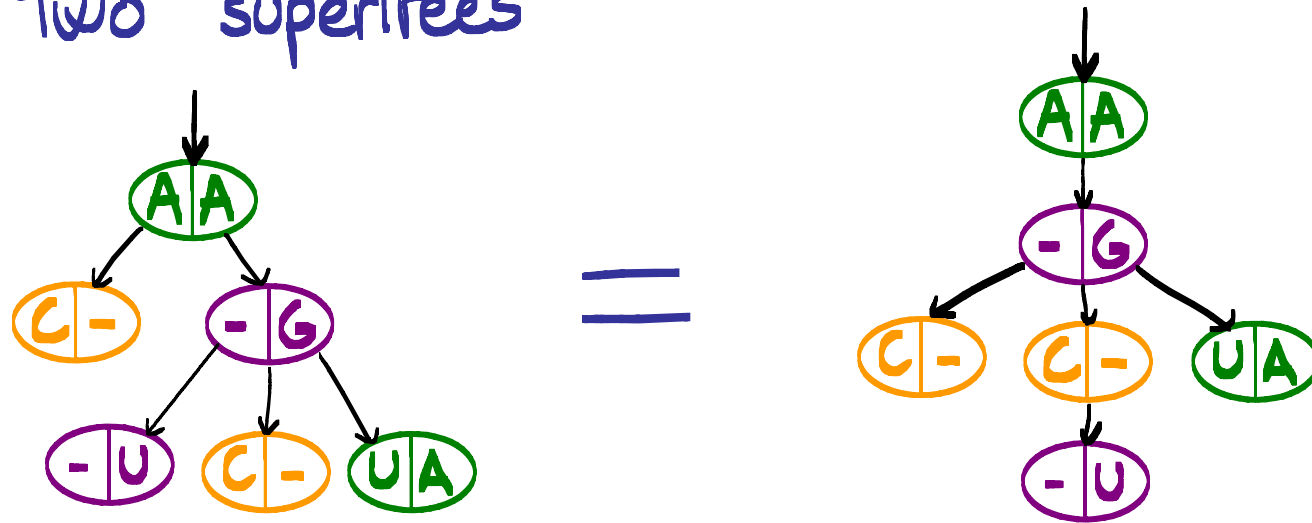


induce the same alignment between the trees

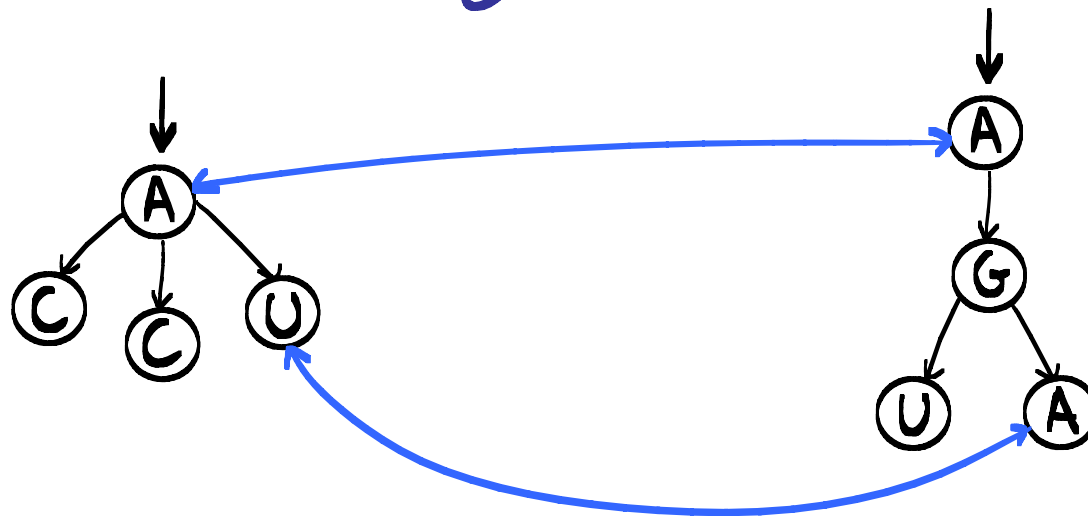


AMBIGUITY OF ALIGNMENTS

The two supertrees



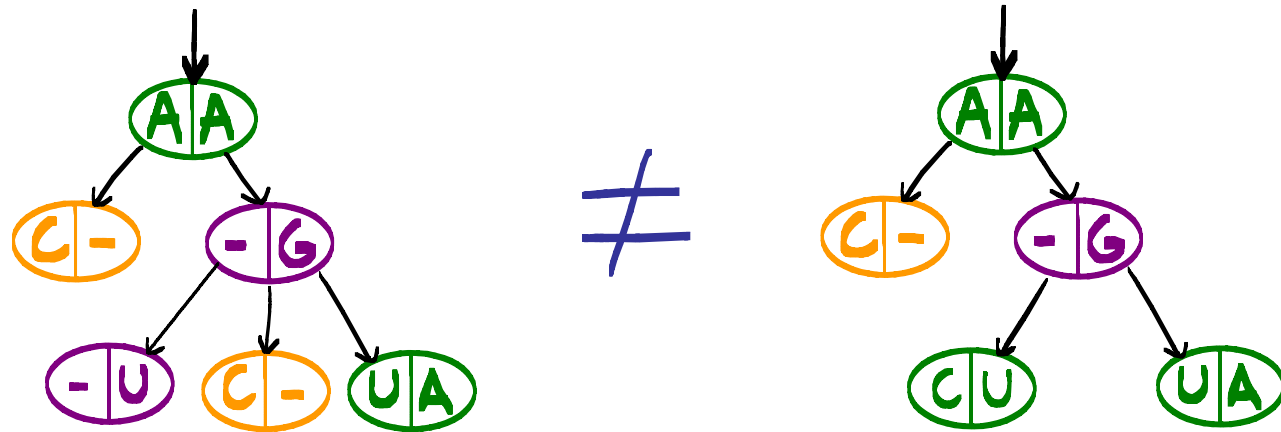
induce the same alignment between the trees



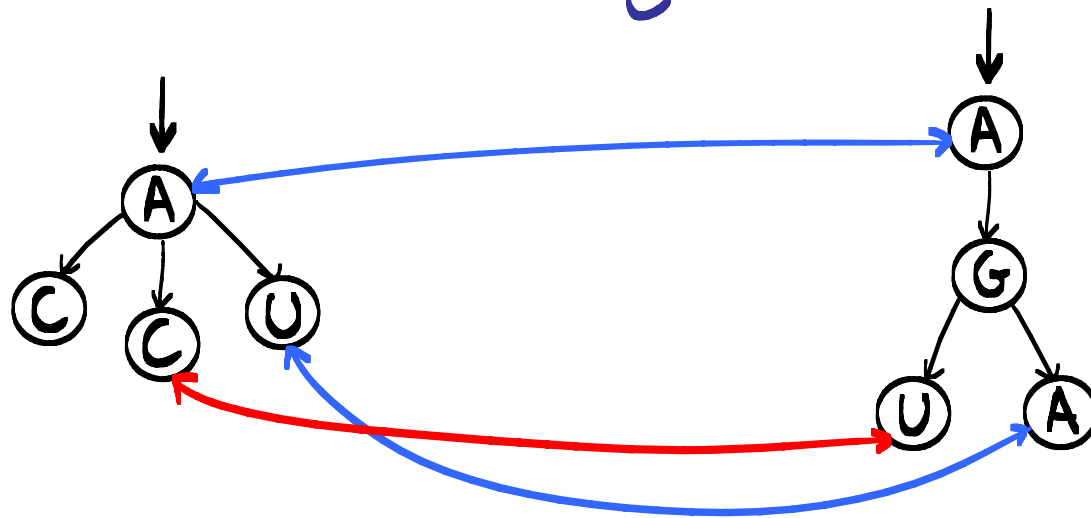
They are the same!

AMBIGUITY OF ALIGNMENTS

The two supertrees



do not induce the same alignment between the trees



A GRAMMAR FOR ALIGNMENTS

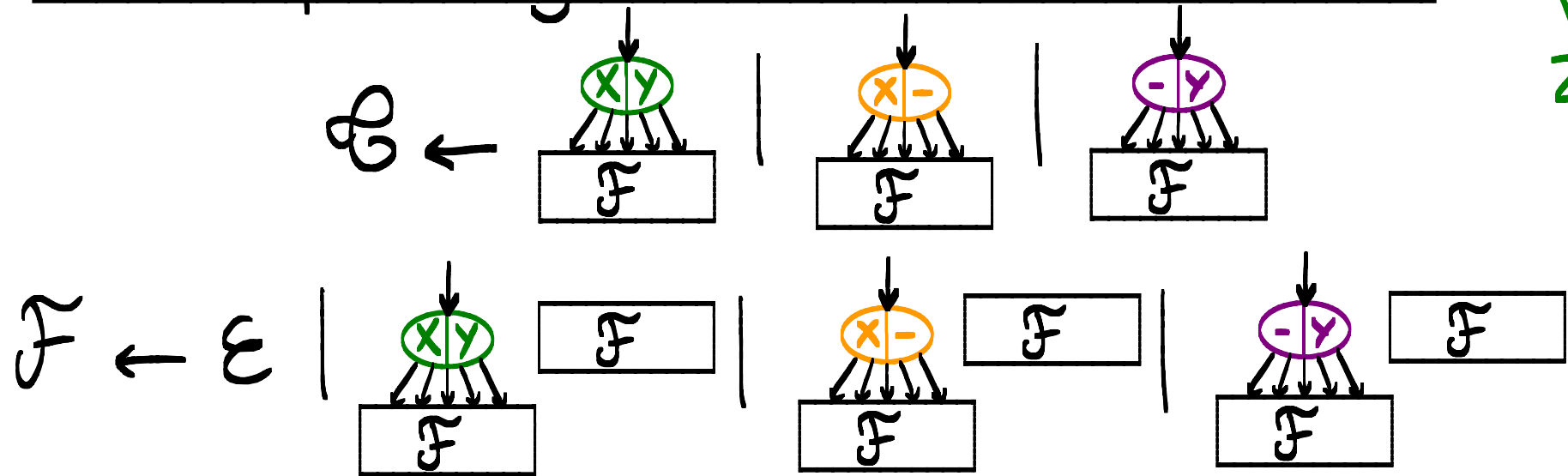
Strategy: Build a context-free grammar that generates every alignment exactly once

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]

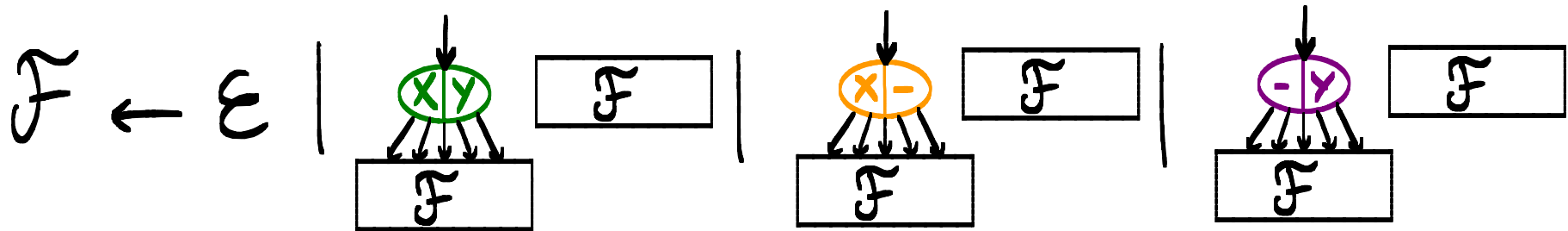
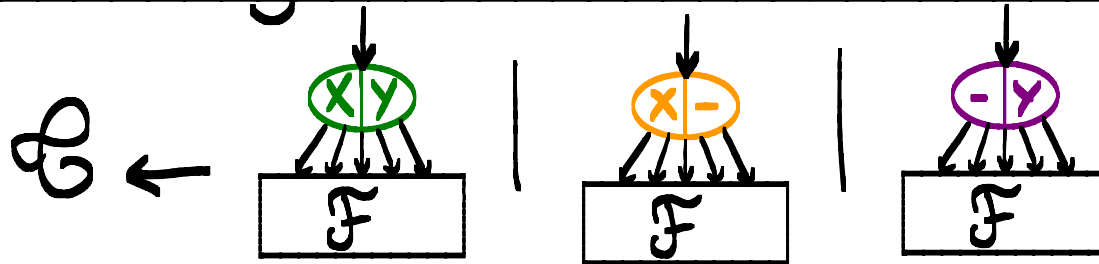


A GRAMMAR FOR ALIGNMENTS

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

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

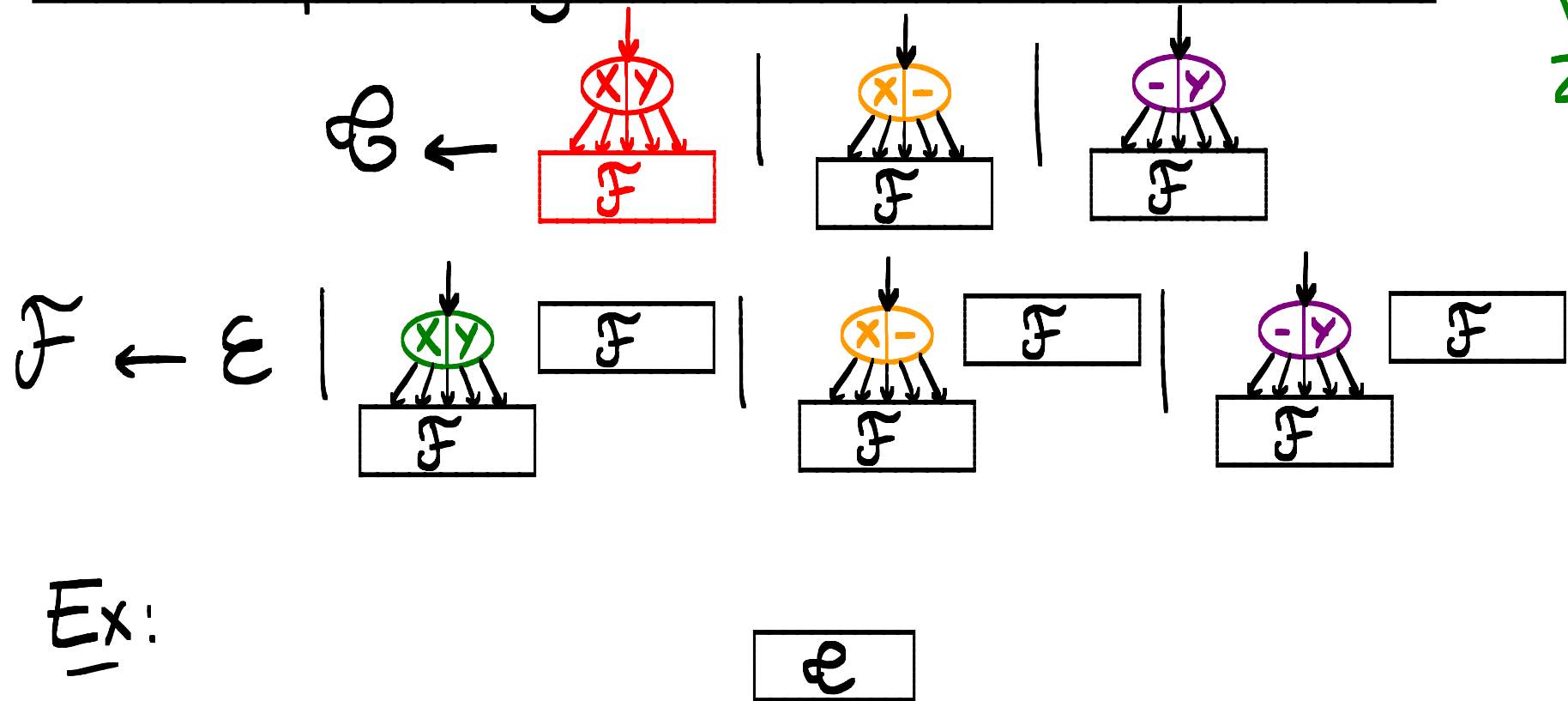


A GRAMMAR FOR ALIGNMENTS

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

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



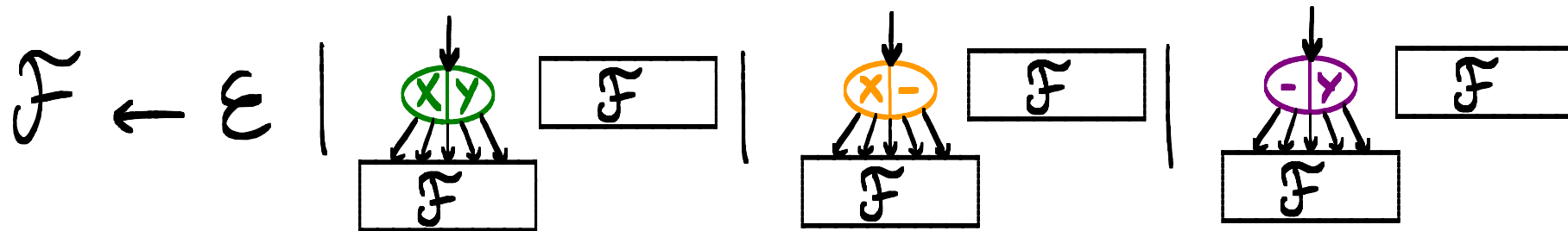
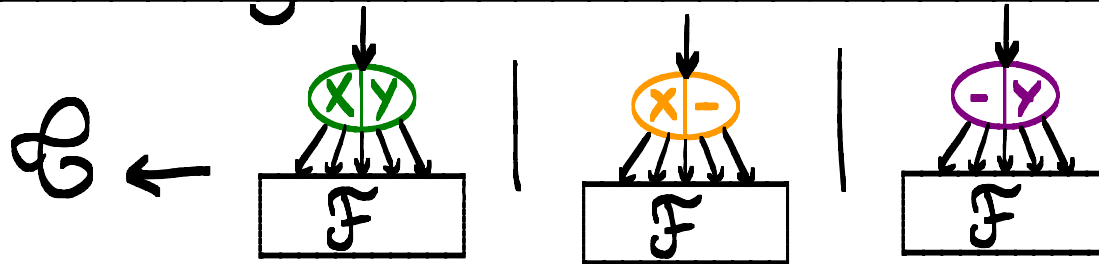
Ex:

A GRAMMAR FOR ALIGNMENTS

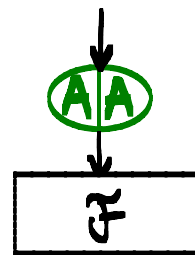
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

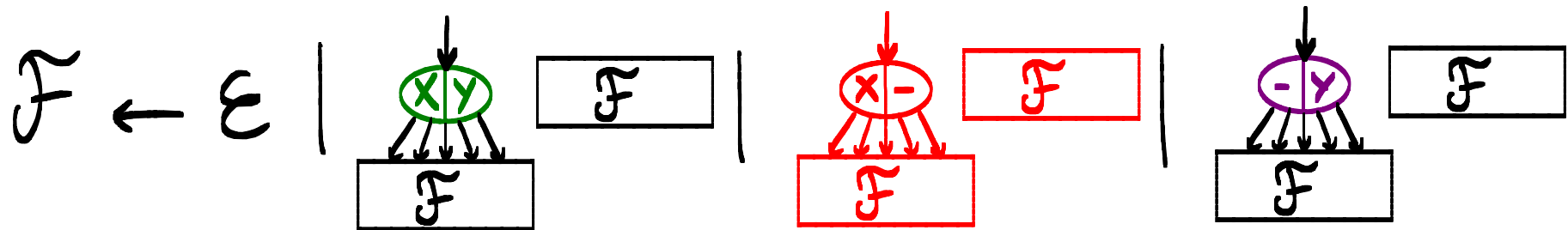
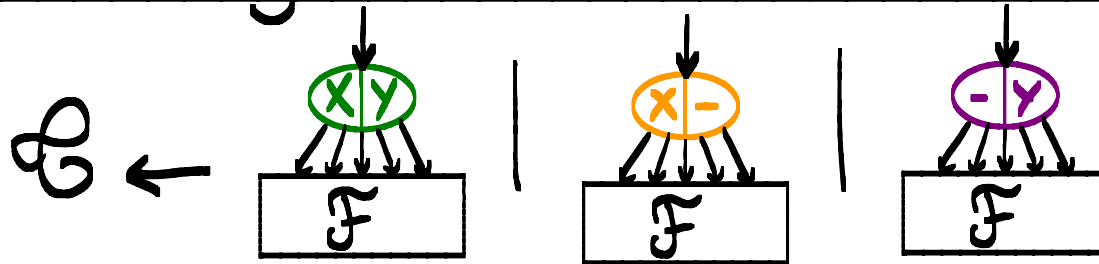


A GRAMMAR FOR ALIGNMENTS

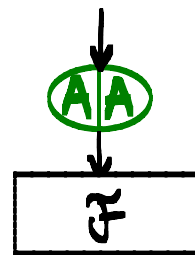
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

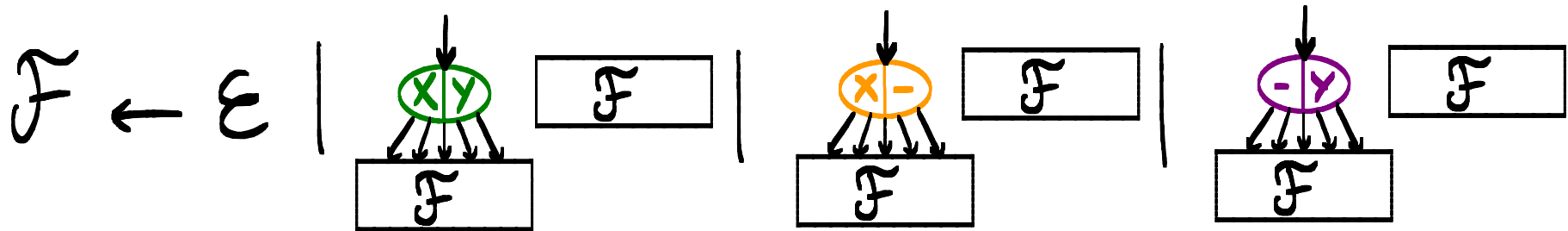
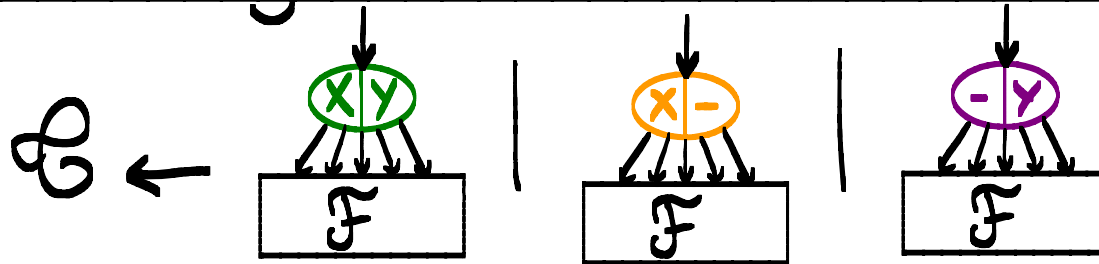


A GRAMMAR FOR ALIGNMENTS

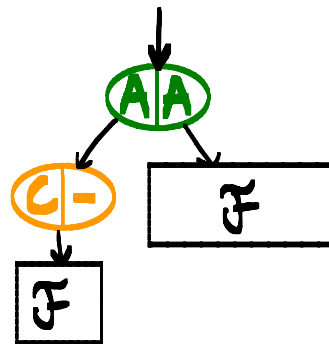
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

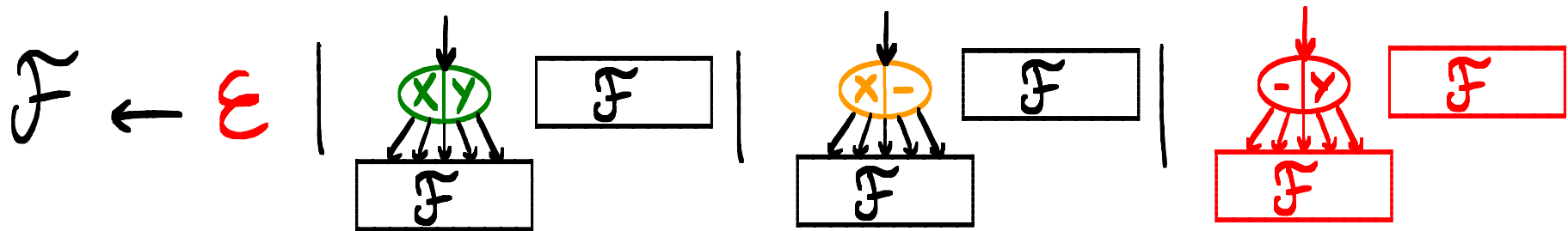
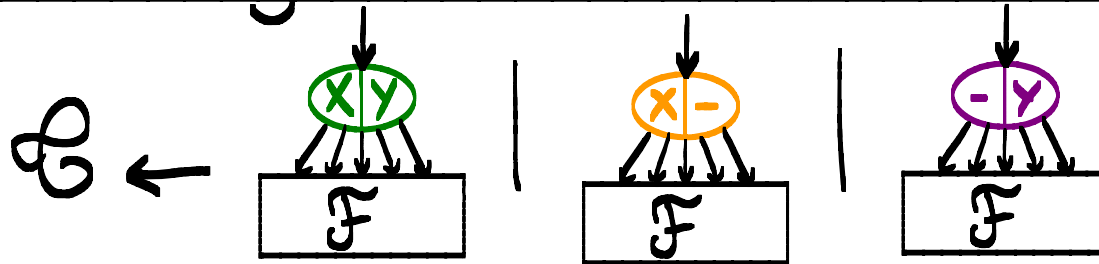


A GRAMMAR FOR ALIGNMENTS

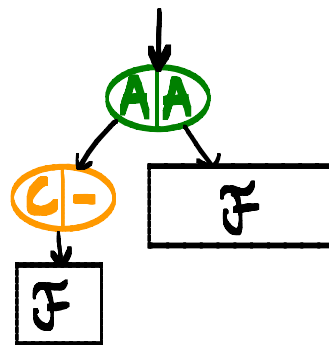
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

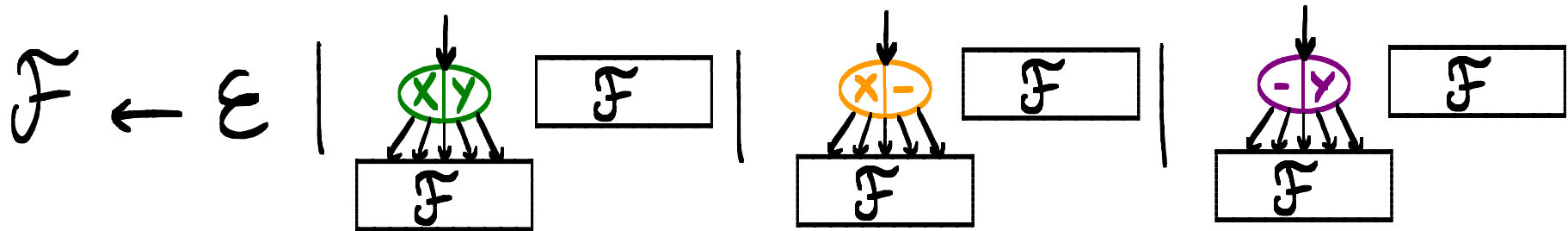
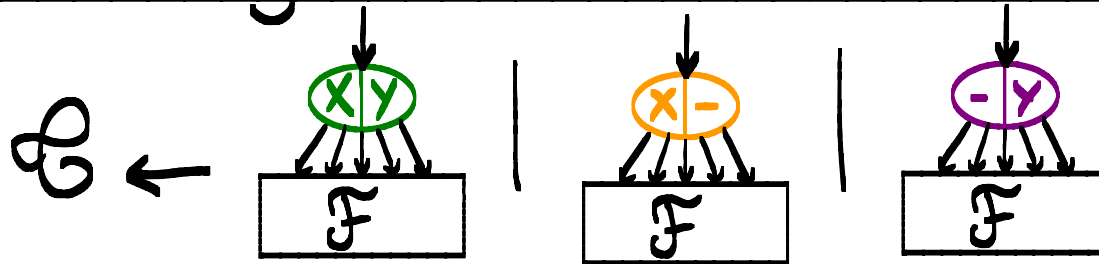


A GRAMMAR FOR ALIGNMENTS

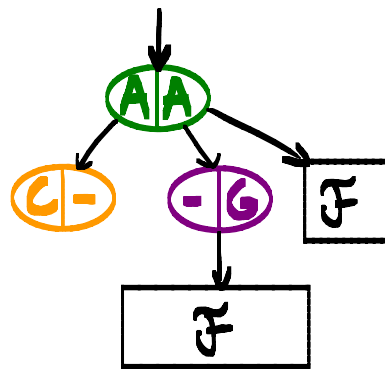
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

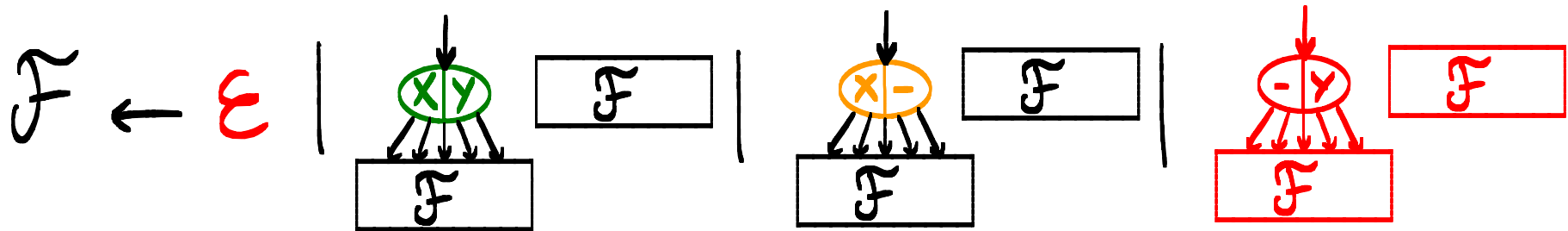
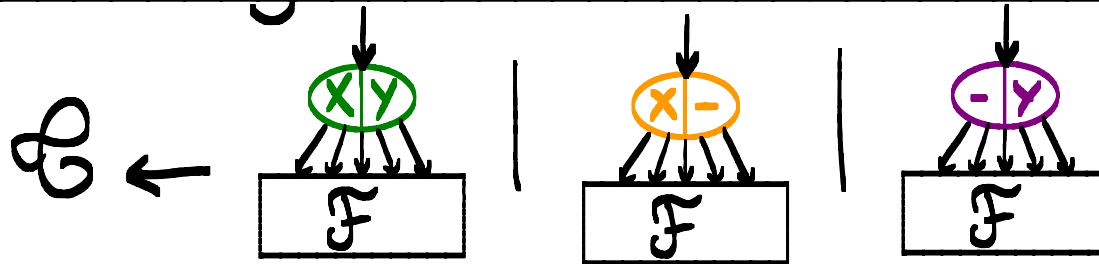


A GRAMMAR FOR ALIGNMENTS

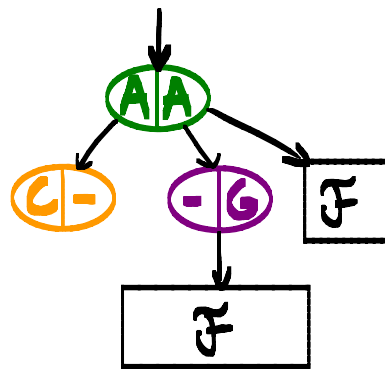
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

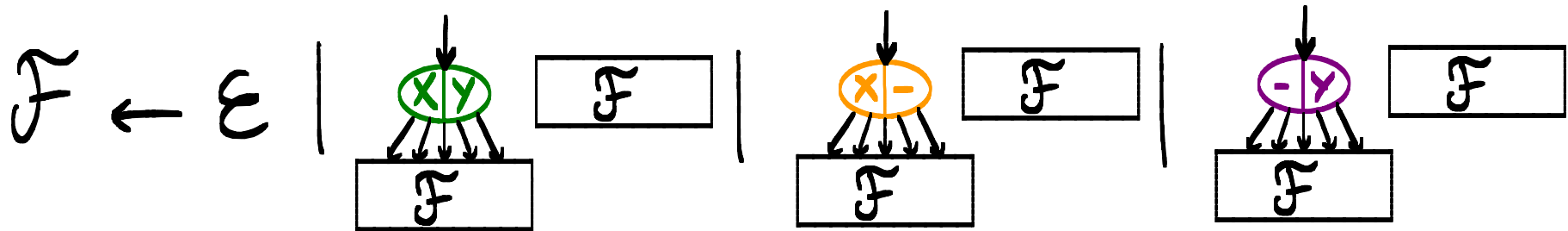
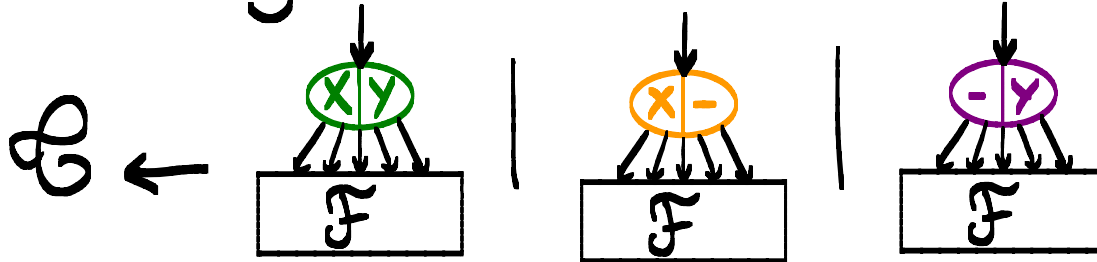


A GRAMMAR FOR ALIGNMENTS

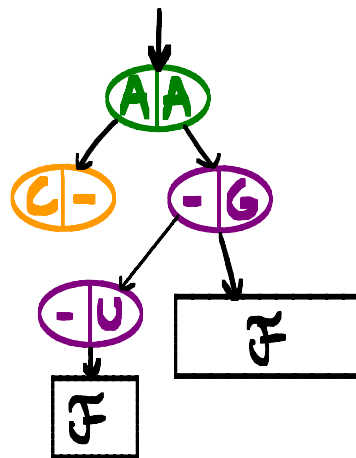
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

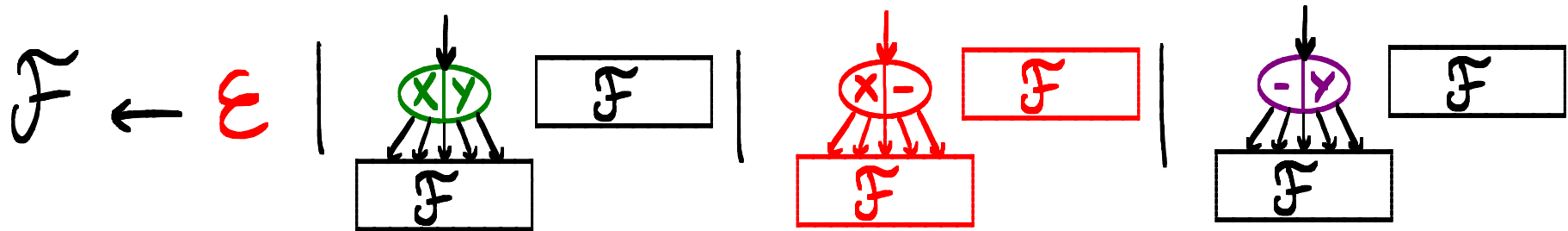
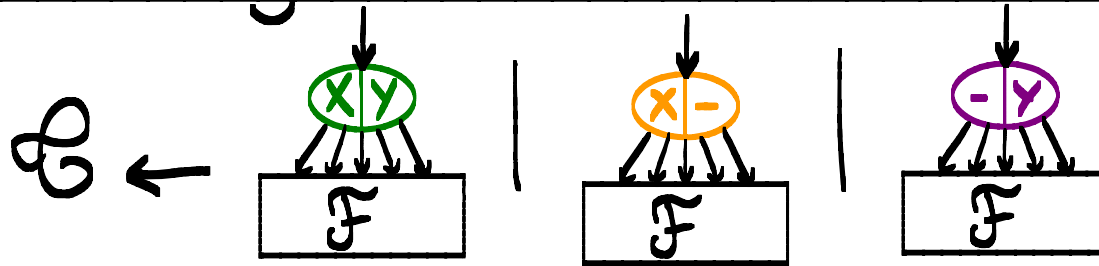


A GRAMMAR FOR ALIGNMENTS

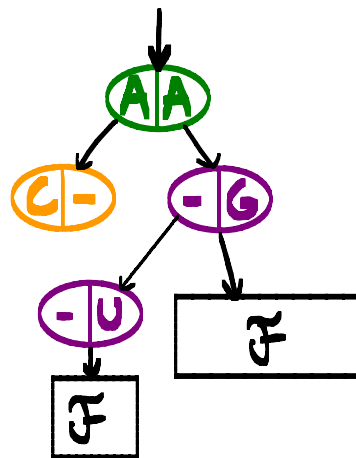
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

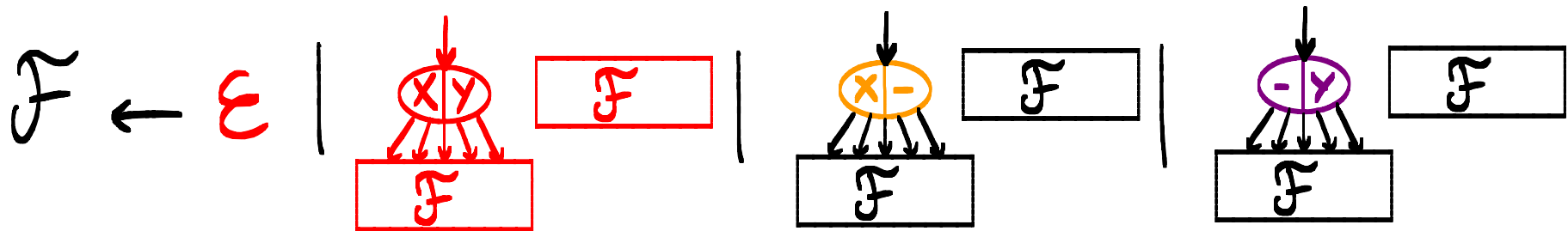
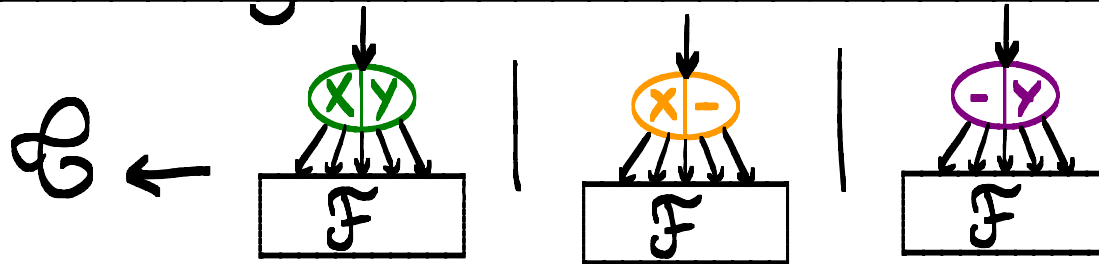


A GRAMMAR FOR ALIGNMENTS

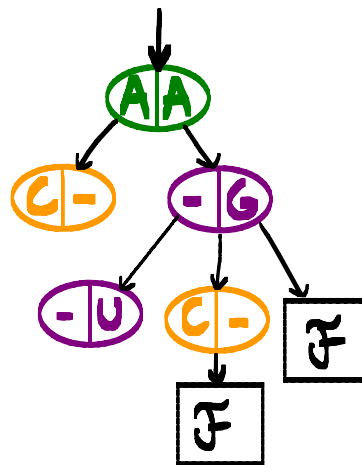
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

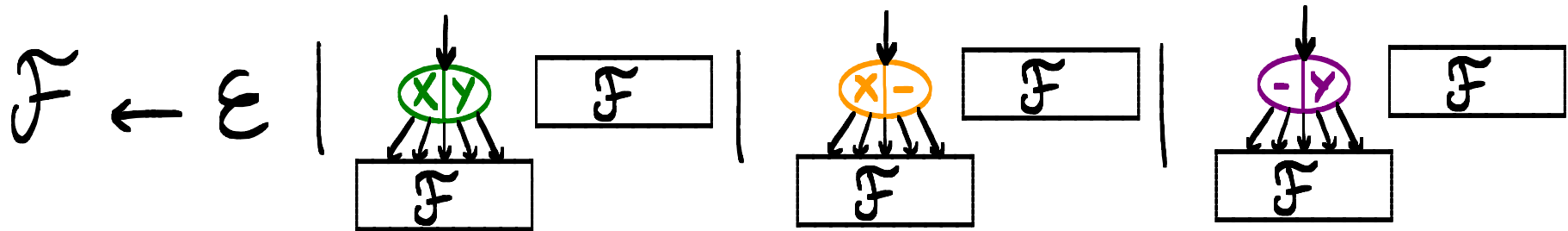
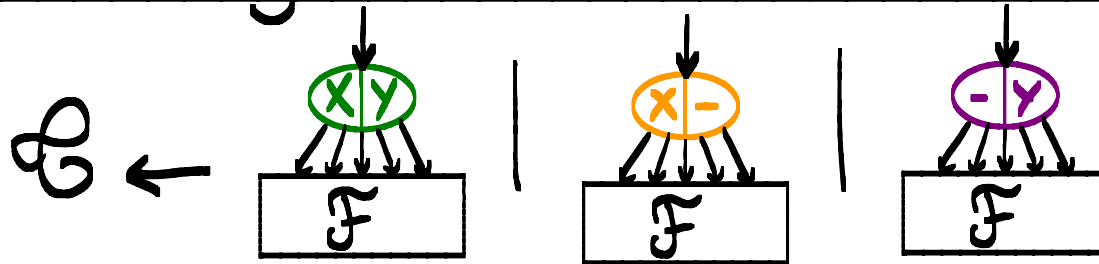


A GRAMMAR FOR ALIGNMENTS

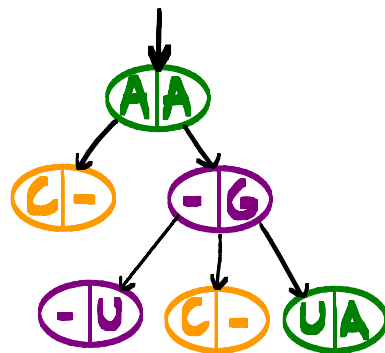
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

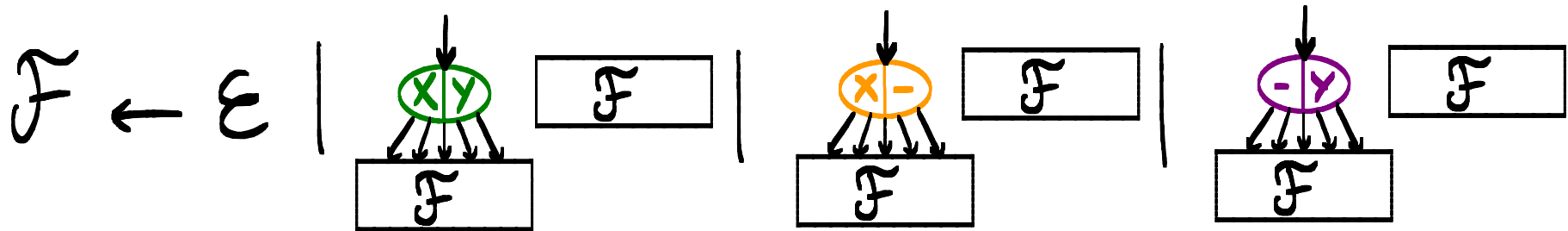
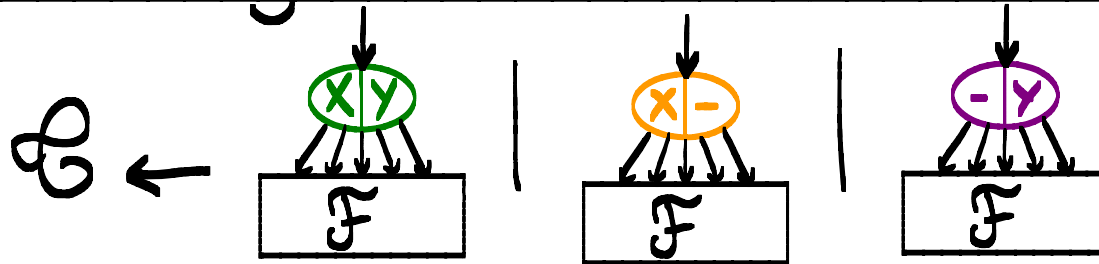


A GRAMMAR FOR ALIGNMENTS

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

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

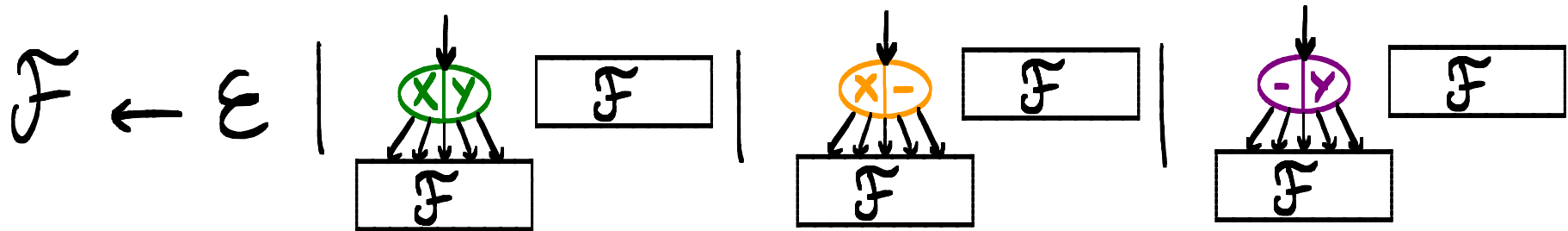
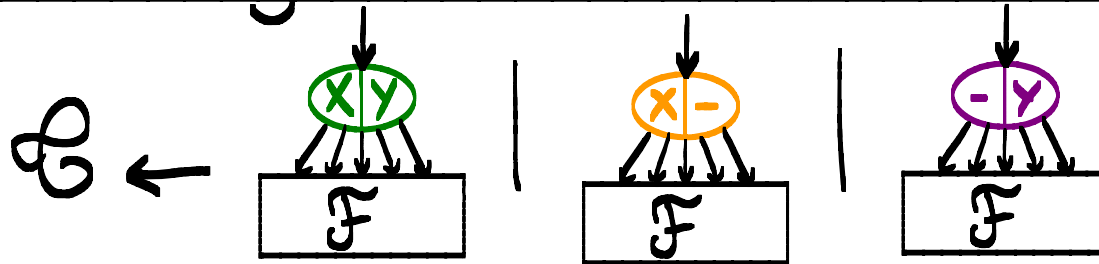


A GRAMMAR FOR ALIGNMENTS

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

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

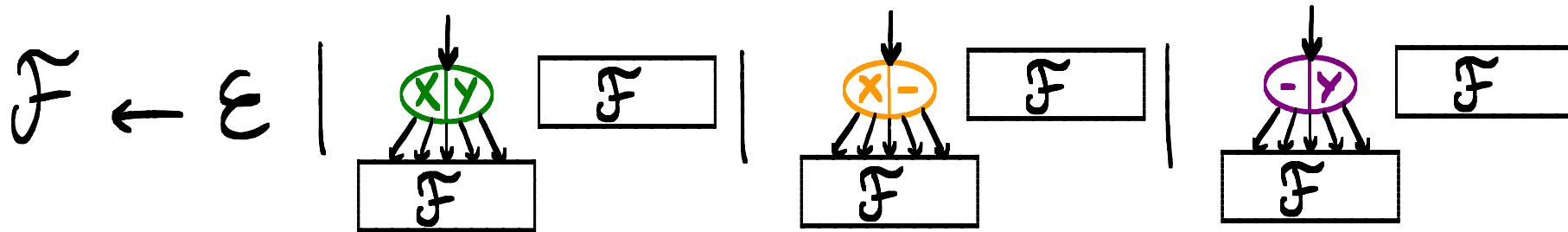
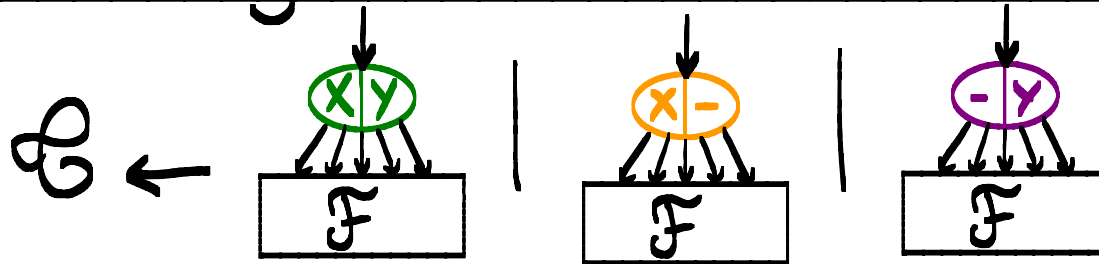


A GRAMMAR FOR ALIGNMENTS

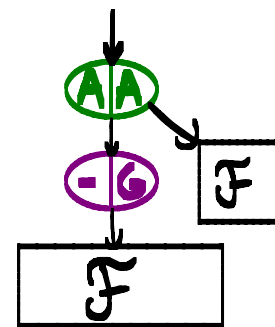
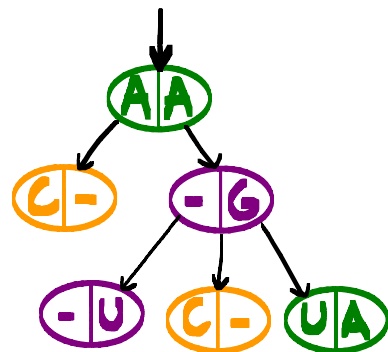
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

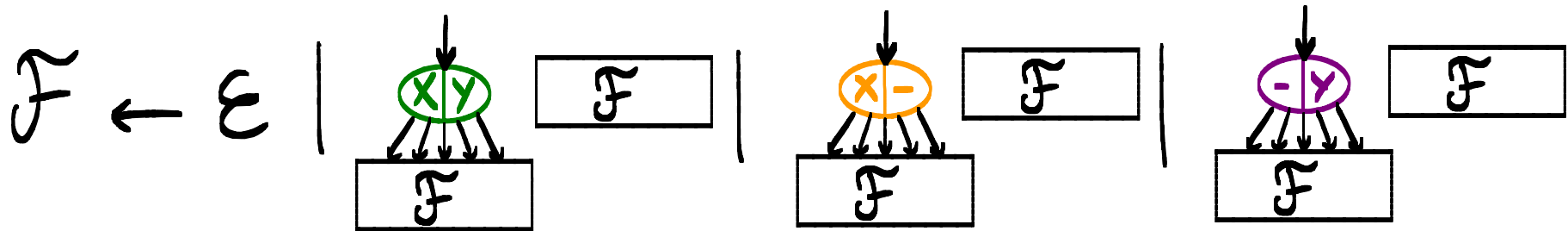
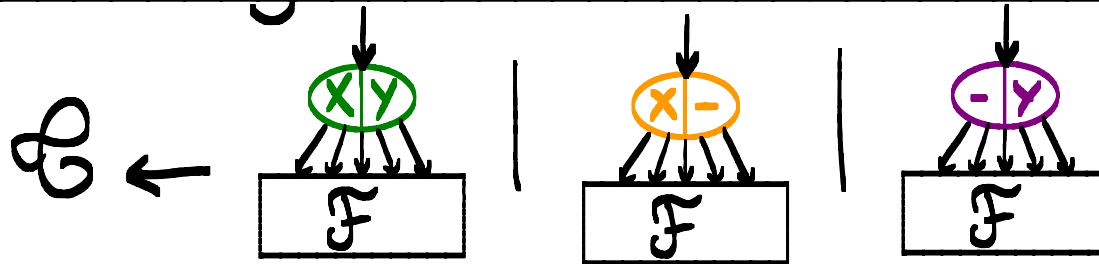


A GRAMMAR FOR ALIGNMENTS

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

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

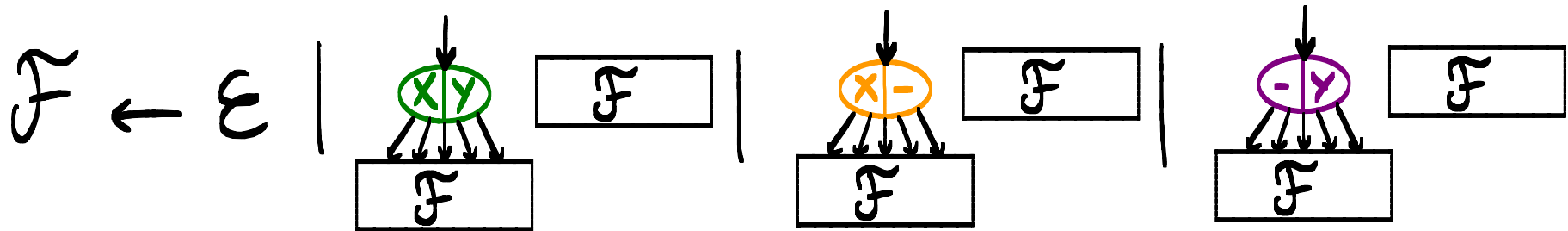
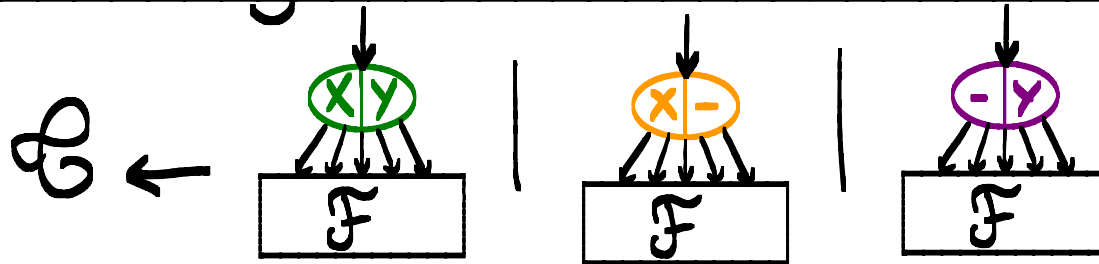


A GRAMMAR FOR ALIGNMENTS

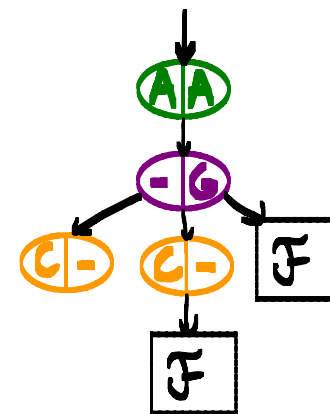
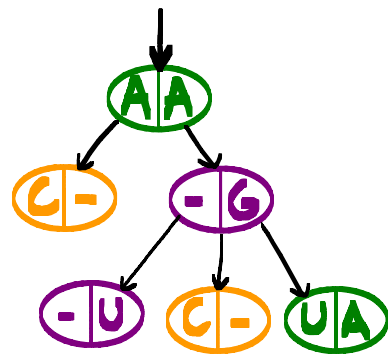
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

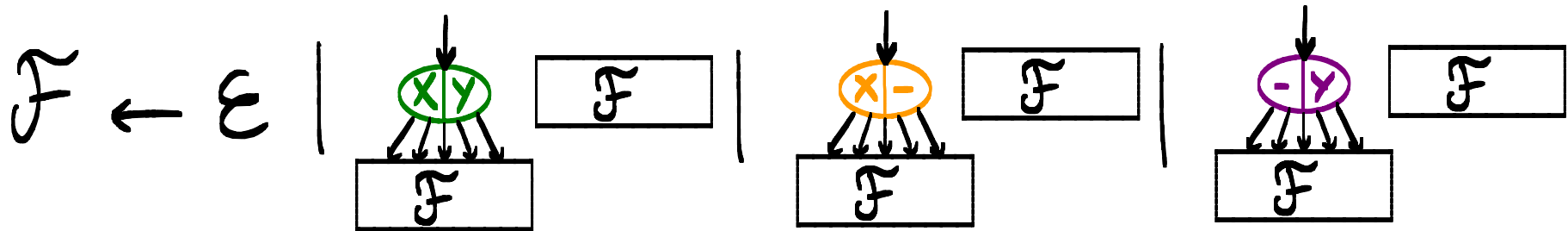
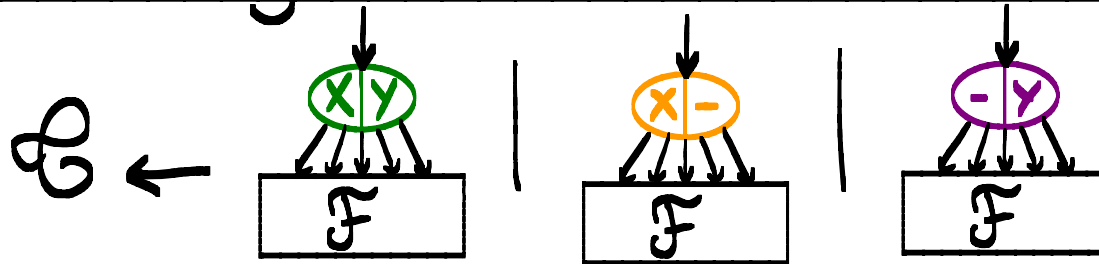


A GRAMMAR FOR ALIGNMENTS

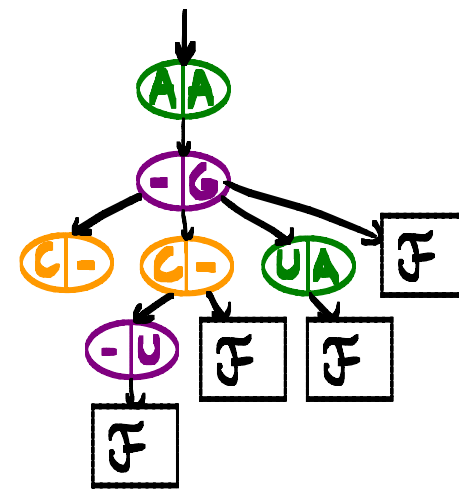
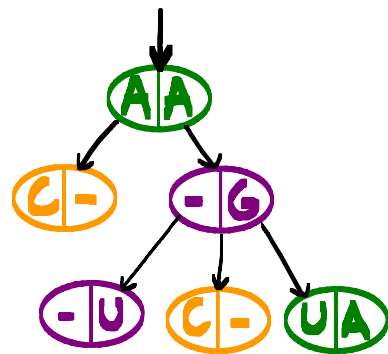
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

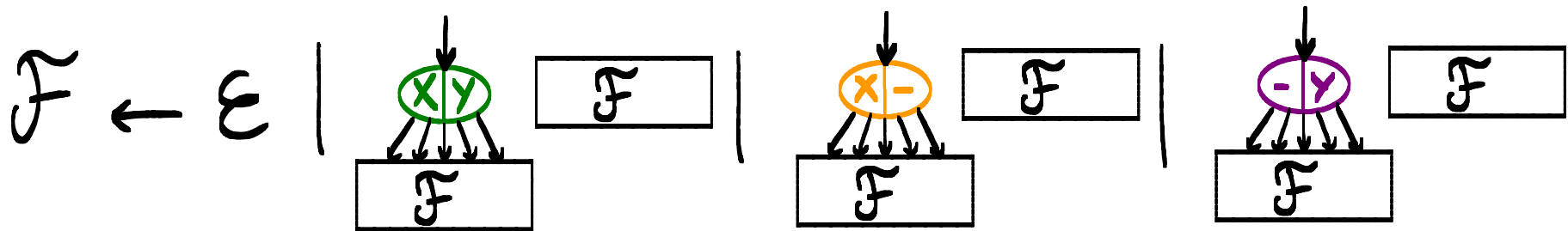
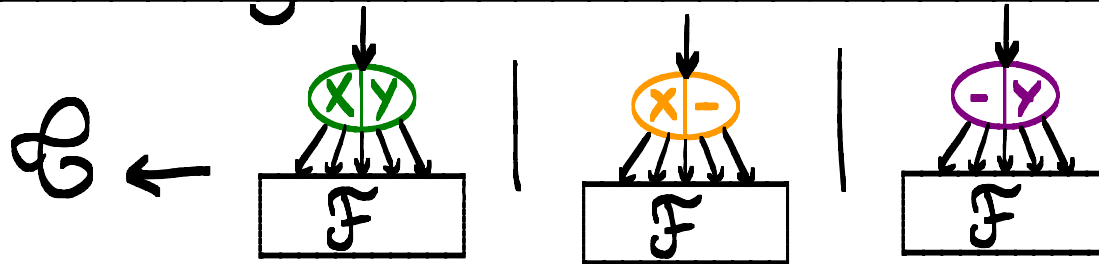


A GRAMMAR FOR ALIGNMENTS

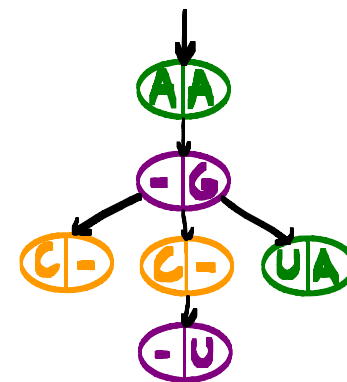
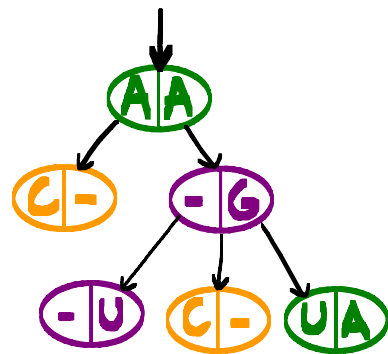
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang,
Wang,
Zhang]



Ex:

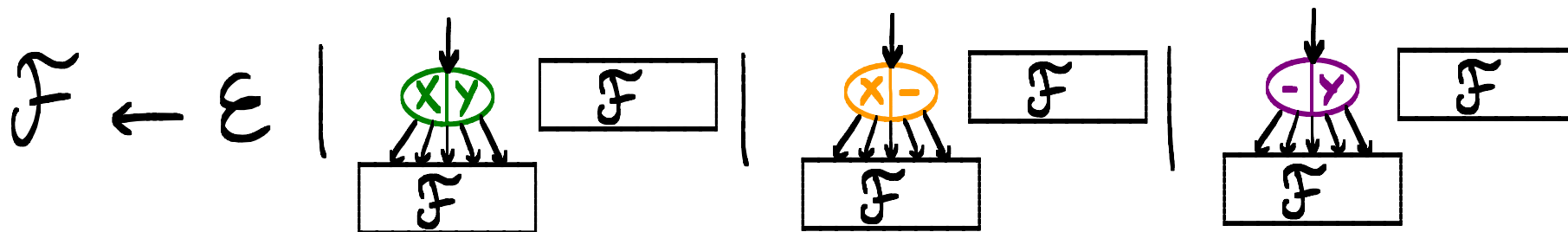
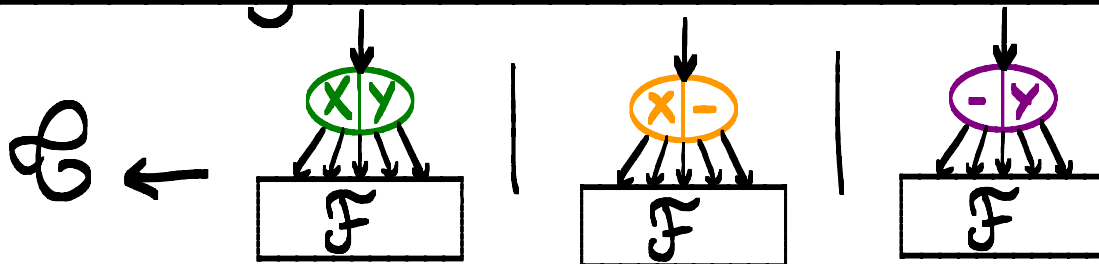


A GRAMMAR FOR ALIGNMENTS

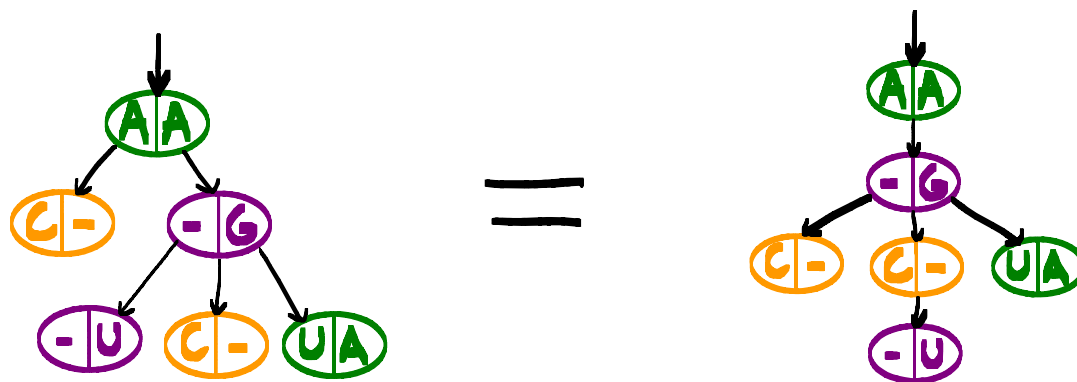
Strategy: Build a context-free grammar that generates every alignment exactly once

An example of grammar that does not work:

[Jiang, Wang, Zhang]



Ex:



ambiguous grammar

A GRAMMAR FOR ALIGNMENTS

Theorem : The set \mathcal{A} generated by the following grammar contains every tree alignment exactly once.

APPLICATION 1: COUNTING.

a_n = number of tree alignments of size n

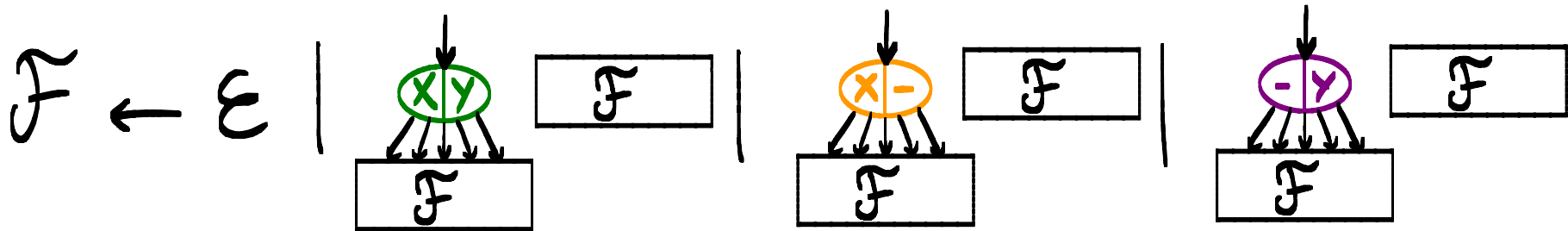
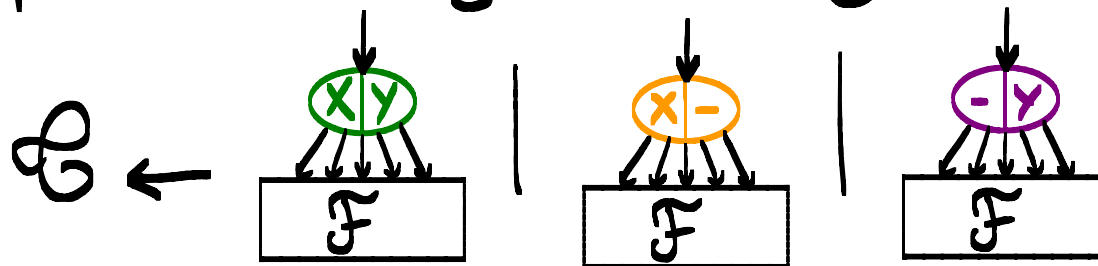
Generating function: $A(z) = \sum_{n \geq 0} a_n z^n$

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:

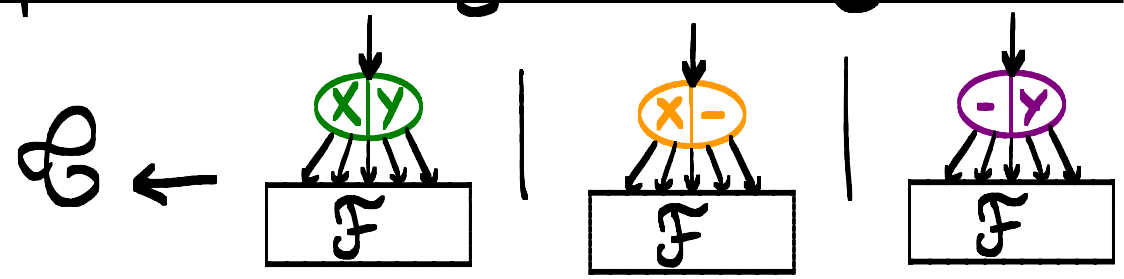


APPLICATION 1: COUNTING.

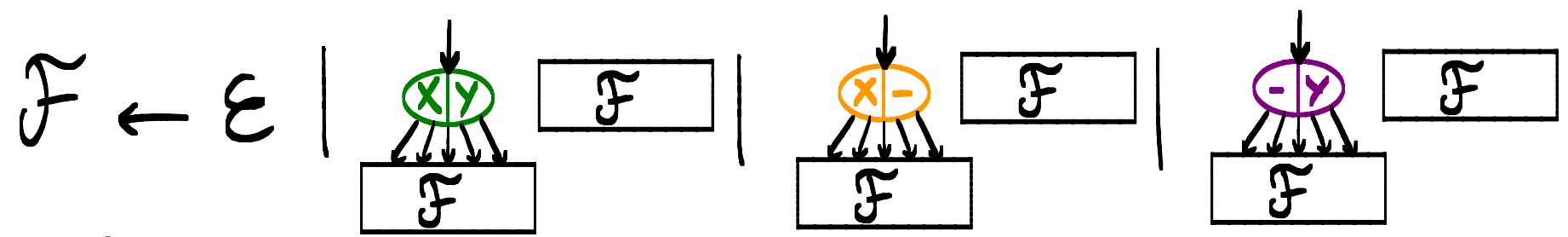
$a_n =$ number of tree alignments of size n

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

The principle on Jiang et al.'s grammar:



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



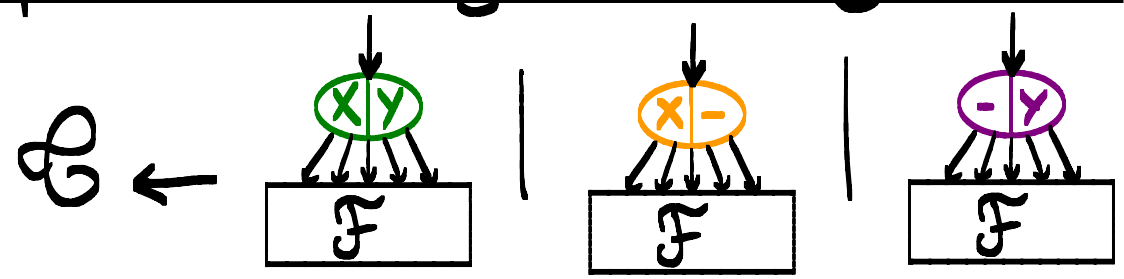
$$F(z) = 1 + F(z) \times F(z) + F(z) \times F(z) + F(z) \times F(z)$$

APPLICATION 1: COUNTING.

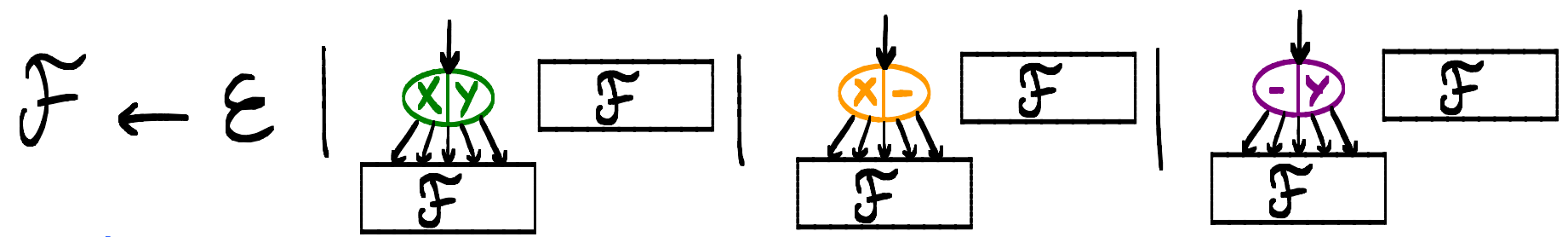
$a_n =$ number of tree alignments of size n

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

The principle on Jiang et al.'s grammar:



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



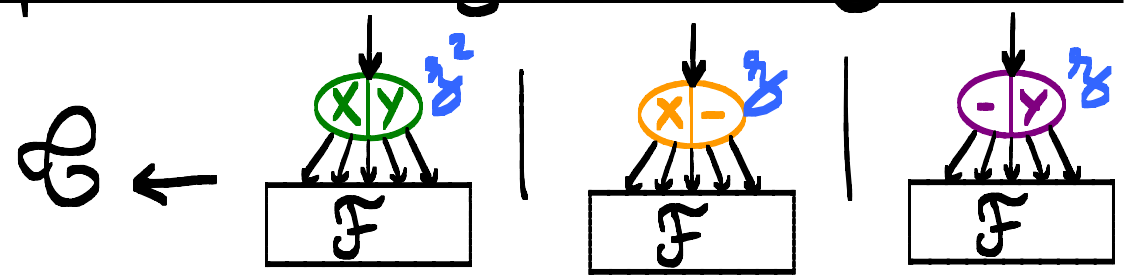
$$F(z) = 1 + F(z)^2 + F(z)^2 + F(z)^2$$

APPLICATION 1: COUNTING.

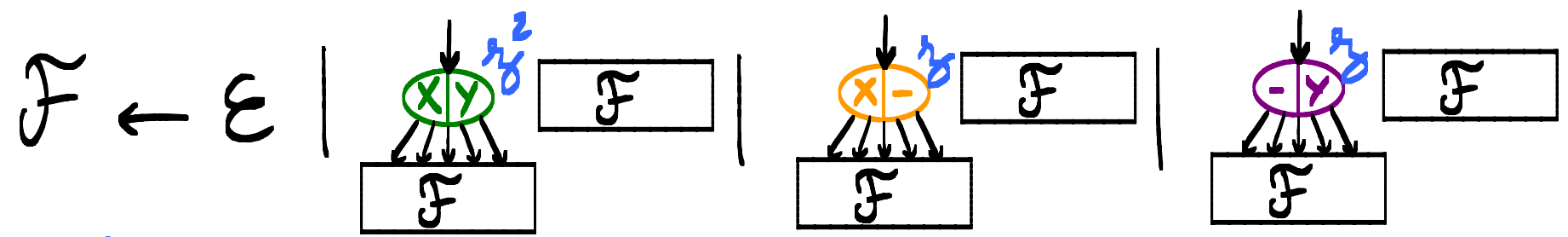
$a_n =$ number of tree alignments of size n

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

The principle on Jiang et al.'s grammar:



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



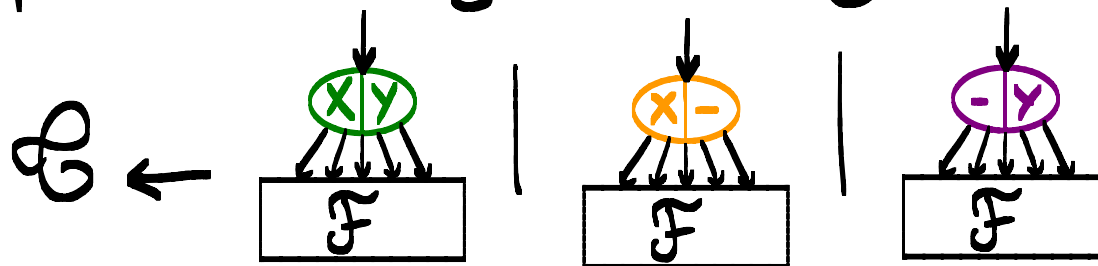
$$F(z) = 1 + F(z)^2 + F(z)^2 + F(z)^2$$

APPLICATION 1: COUNTING.

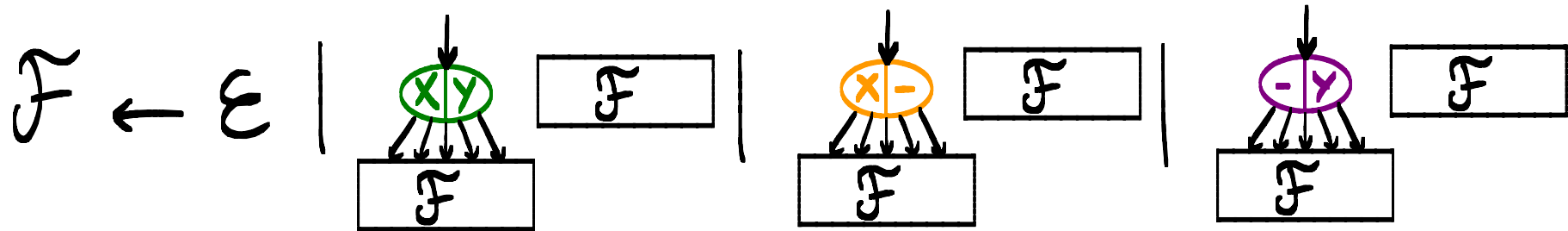
$a_n =$ number of tree alignments of size n

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

The principle on Jiang et al.'s grammar:



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



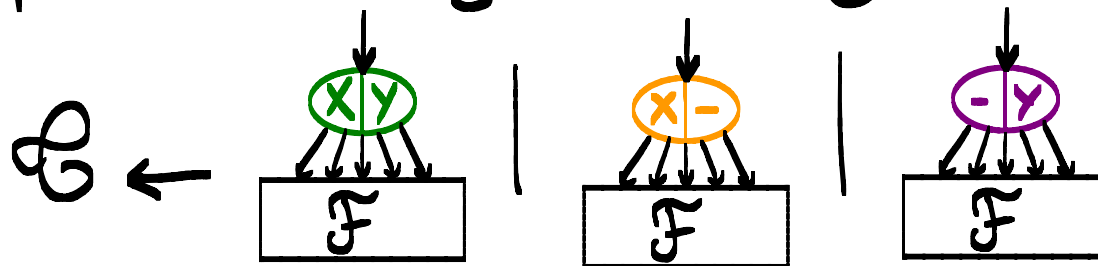
$$F(z) = 1 + z^2 \times F(z)^2 + z \times F(z)^2 + z \times F(z)^2$$

APPLICATION 1: COUNTING.

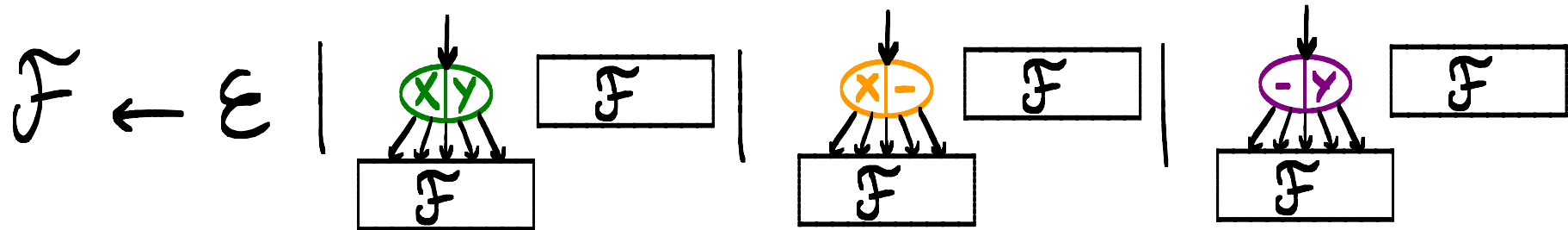
$a_{n,k}$ = number of tree alignments of size n
and k matches

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

The principle on Jiang et al.'s grammar:



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



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

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

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

and

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

SOME STATISTICAL PROPERTIES

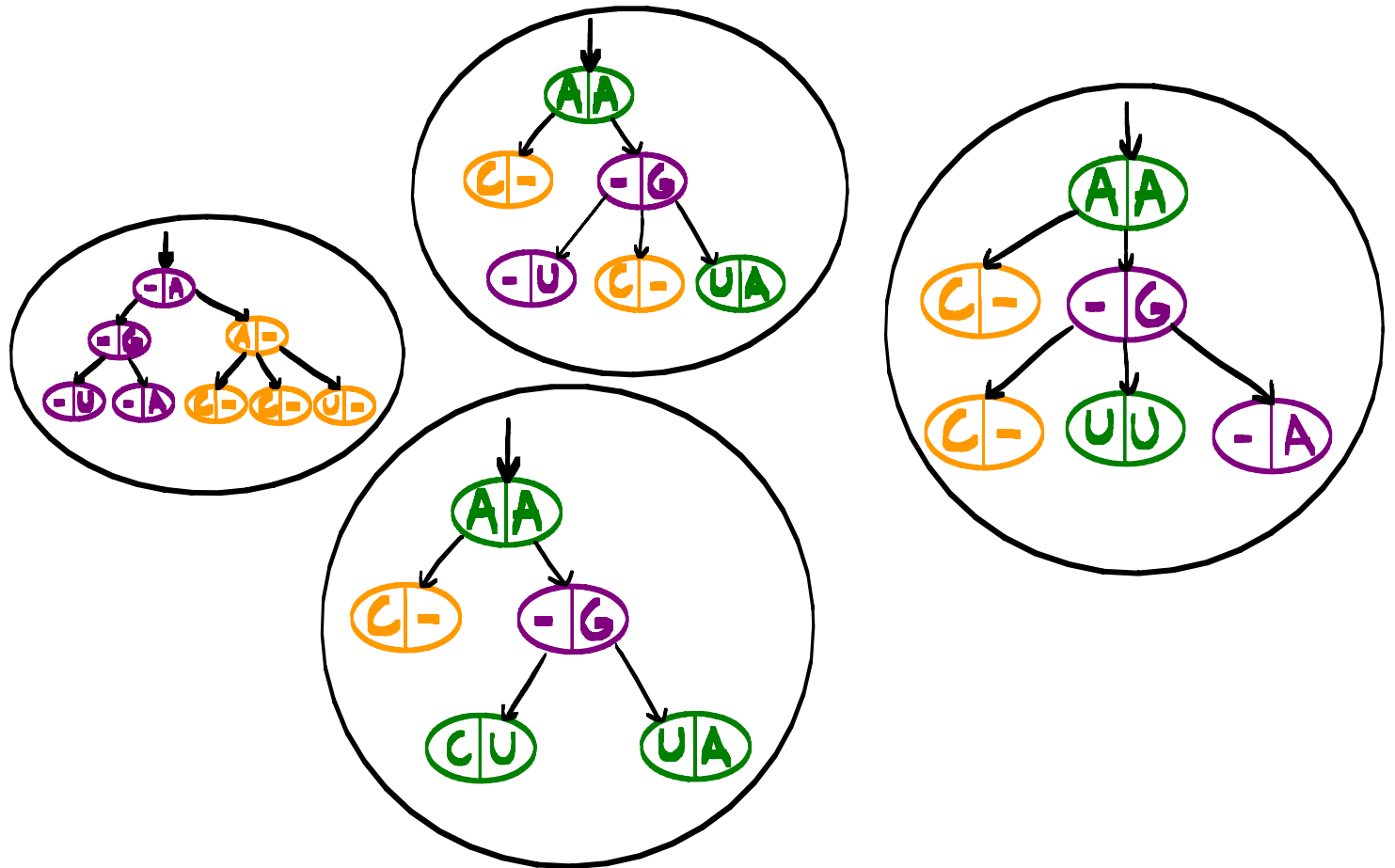
Theorem There are on average
 $C \times 1.5^n$ alignments
between two random trees of cumulative size n
where $C = 0.299\dots$

Corollary: 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
 $\propto e^{-\frac{\text{cost}(A)}{k}}$
(Gibbs-Boltzmann
distribution)



APPLICATION 2. SAMPLING

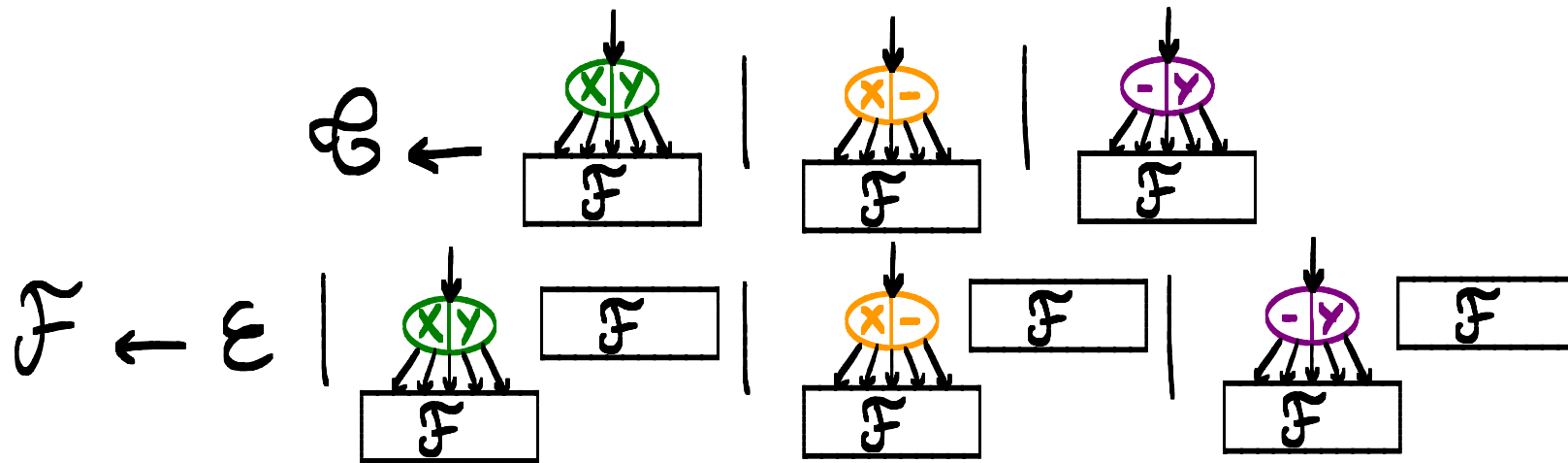
Objective: Sampling alignments under the Gibbs-Boltzmann probability distribution.

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:



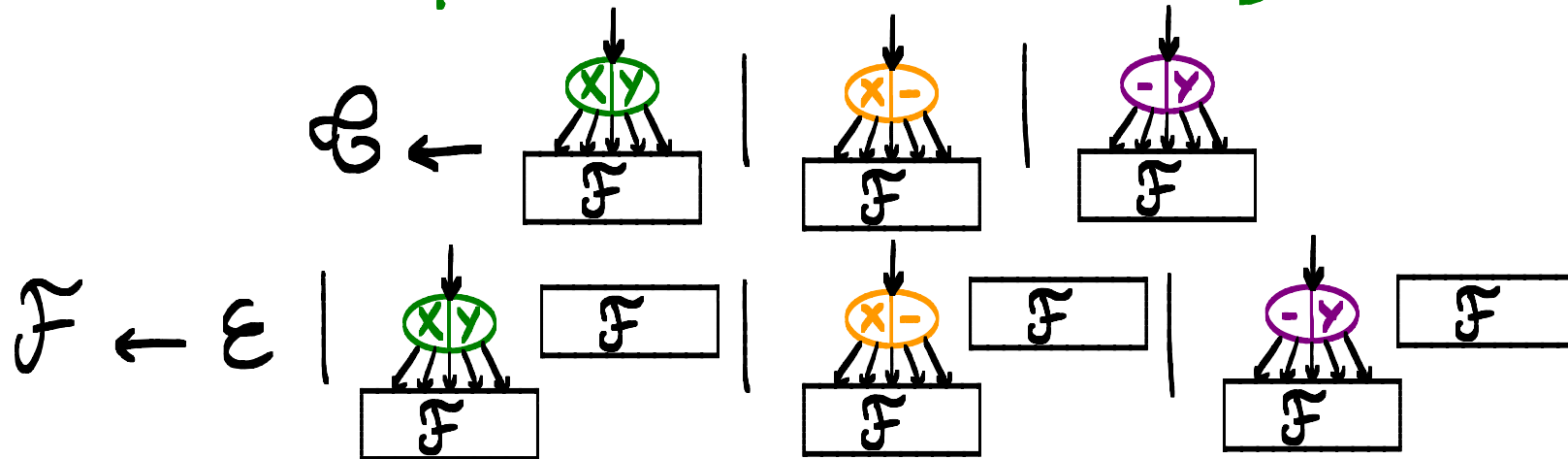
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

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

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



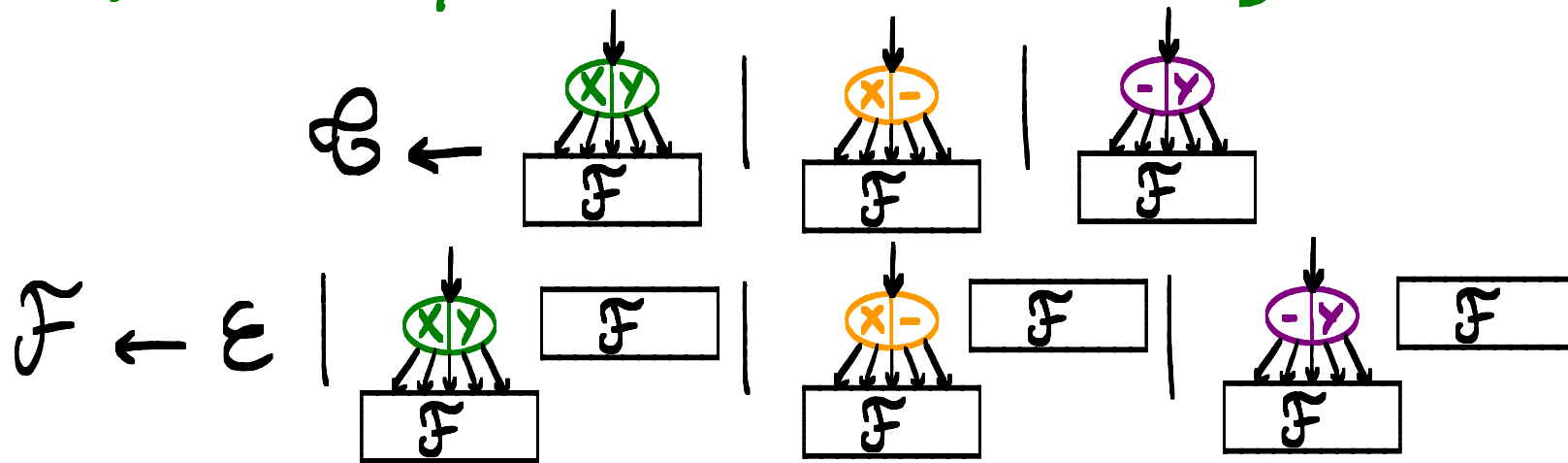
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

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

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

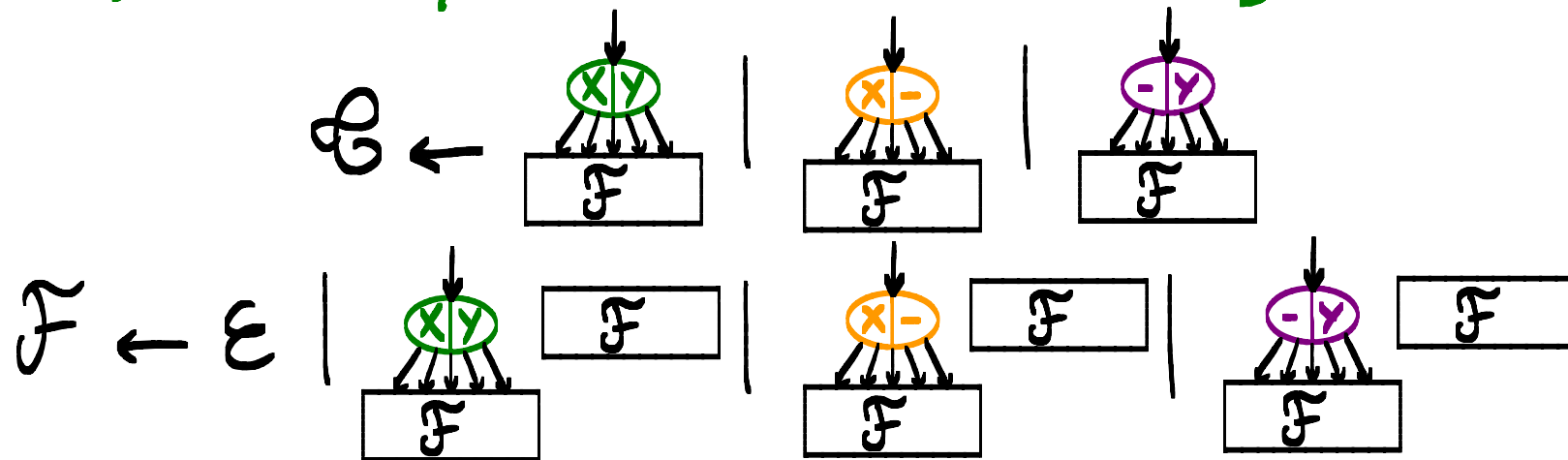
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

The principle on Jiang et al.'s grammar:

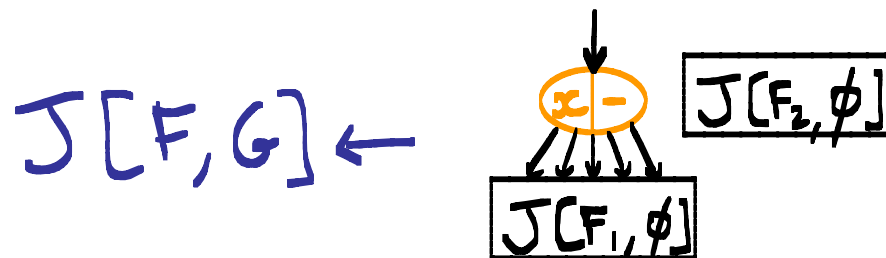
We fix two trees S and T .

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

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



If $F = \begin{matrix} \text{Ⓢ} \\ \boxed{F_1} \end{matrix} \boxed{F_2}$ and $G = \emptyset$, then



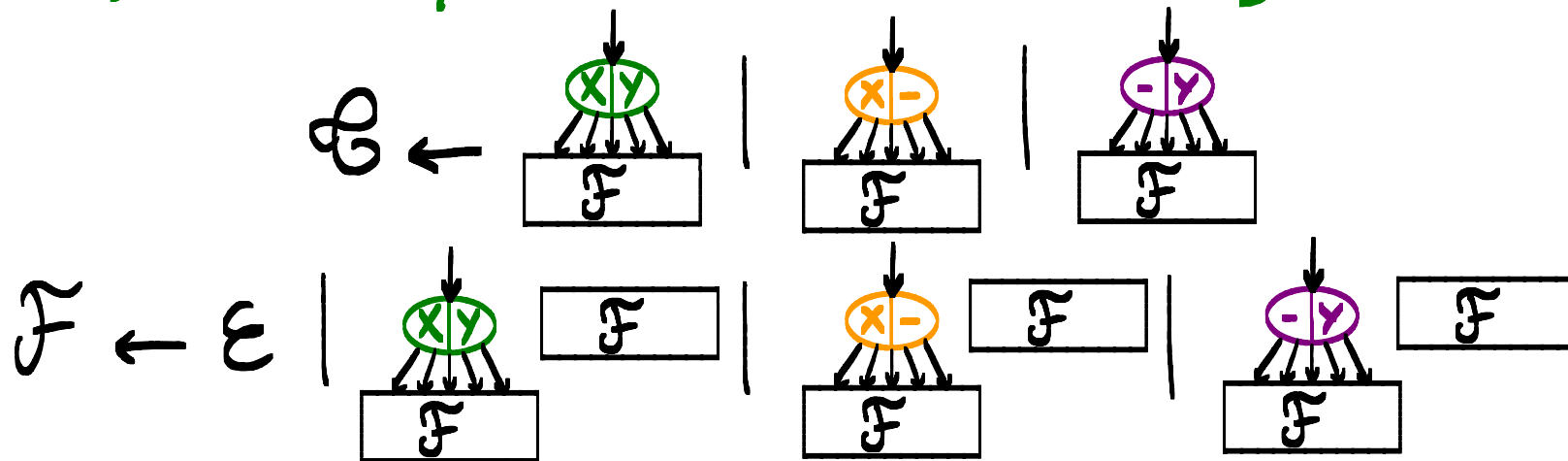
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

The principle on Jiang et al.'s grammar:

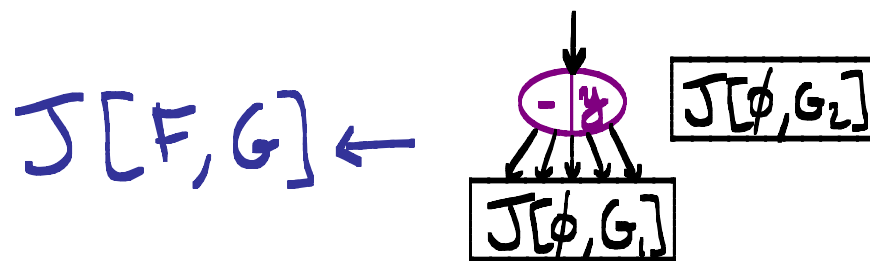
We fix two trees S and T .

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 $G = \begin{matrix} \text{root} \\ \swarrow \quad \searrow \\ G_1 \quad G_2 \end{matrix}$, then



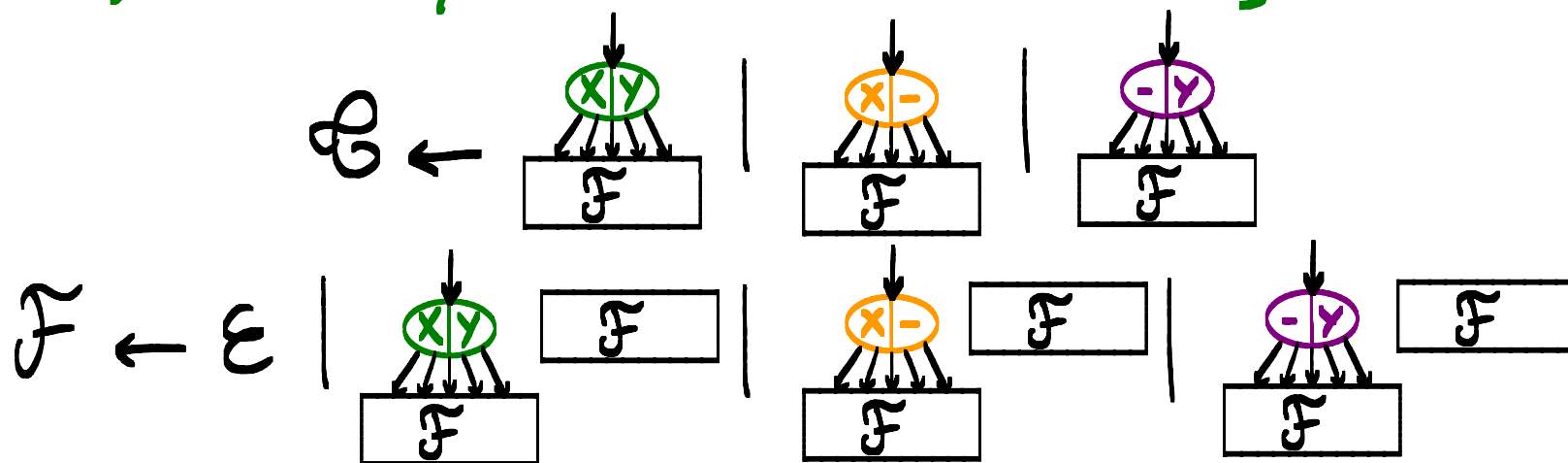
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

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

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



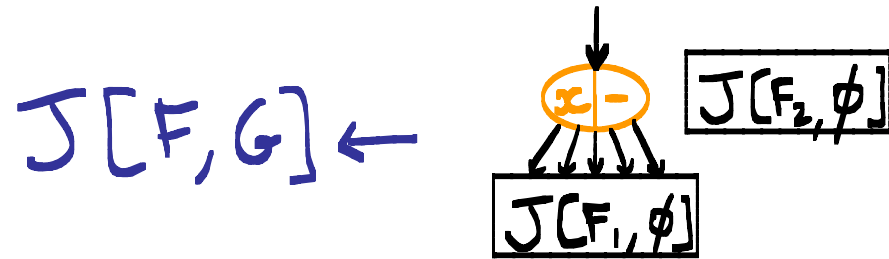
If $F = \begin{matrix} \text{root} \\ \swarrow \searrow \\ F_1 \quad F_2 \end{matrix}$ and $G = \begin{matrix} \text{root} \\ \swarrow \searrow \\ G_1 \quad G_2 \end{matrix}$, then



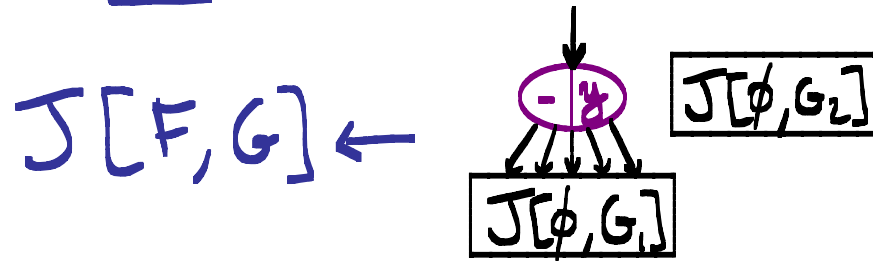
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

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

• If $F = \begin{array}{c} \textcircled{z} \\ \text{---} \\ \boxed{F_1} \end{array} \boxed{F_2}$ and $G = \emptyset$, then



• If $F = \emptyset$ and $G = \begin{array}{c} \textcircled{y} \\ \text{---} \\ \boxed{G_1} \end{array} \boxed{G_2}$, then



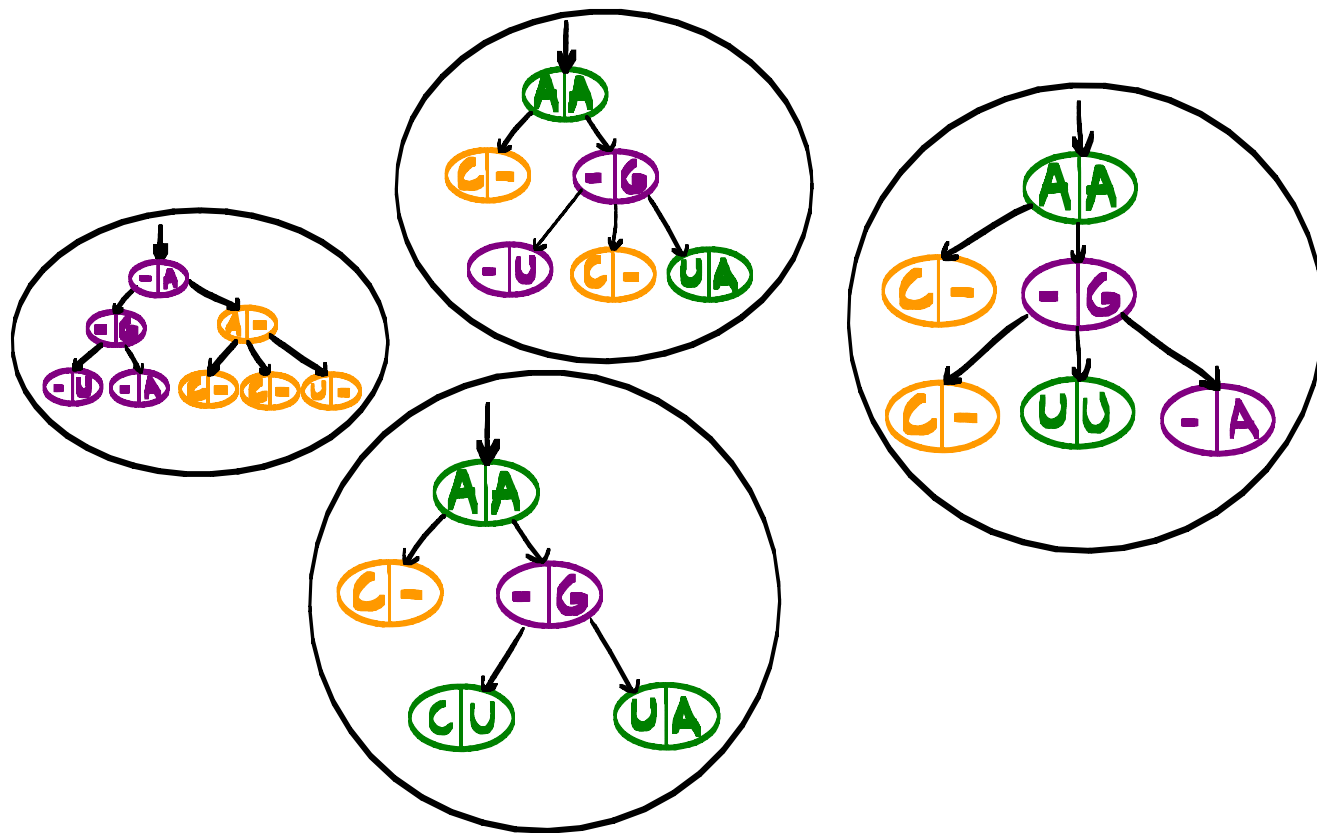
• If $F = \begin{array}{c} \textcircled{z} \\ \text{---} \\ \boxed{F_1} \end{array} \boxed{F_2}$ and $G = \begin{array}{c} \textcircled{y} \\ \text{---} \\ \boxed{G_1} \end{array} \boxed{G_2}$, then



GIBBS-BOLTZMANN DISTRIBUTION

$$P(\text{alignment } A \text{ between } S \text{ and } T) = \frac{e^{-\frac{\text{cost}(A)}{k}}}{Z_{S,T}}$$

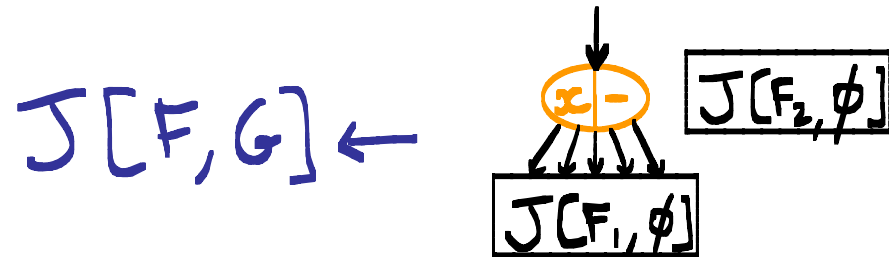
$$Z_{S,T} = \sum_{A' \text{ alignment between } S \text{ and } T} e^{-\frac{\text{cost}(A')}{k}}$$



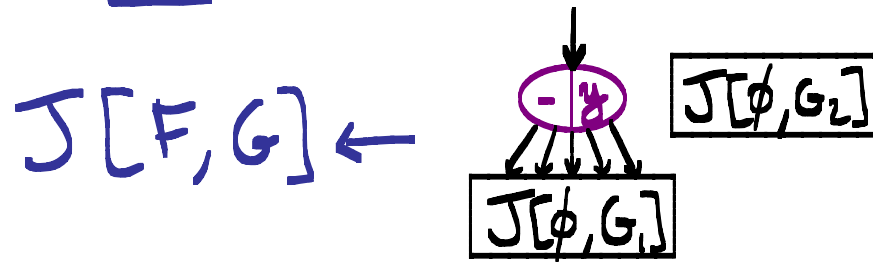
GRAMMAR OF ALIGNMENTS BETWEEN TWO FIXED TREES

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

• If $F = \begin{array}{c} \textcircled{z} \\ \text{---} \\ \boxed{F_1} \end{array} \boxed{F_2}$ and $G = \emptyset$, then



• If $F = \emptyset$ and $G = \begin{array}{c} \textcircled{y} \\ \text{---} \\ \boxed{G_1} \end{array} \boxed{G_2}$, then



• If $F = \begin{array}{c} \textcircled{z} \\ \text{---} \\ \boxed{F_1} \end{array} \boxed{F_2}$ and $G = \begin{array}{c} \textcircled{y} \\ \text{---} \\ \boxed{G_1} \end{array} \boxed{G_2}$, then



PARTITION FUNCTION

• If $F = \emptyset$ and $G = \emptyset$, then $Z_{F,G} \leftarrow 0$

• If $F = \begin{array}{c} \textcircled{2} \\ \text{---} \\ \boxed{F_1} \end{array} \boxed{F_2}$ and $G = \emptyset$, then

$$Z_{F,G} \leftarrow e^{-\frac{1}{K}} \times Z_{F_1, \emptyset} \times Z_{F_2, \emptyset}$$

• If $F = \emptyset$ and $G = \begin{array}{c} \textcircled{2} \\ \text{---} \\ \boxed{G_1} \end{array} \boxed{G_2}$, then

$$Z_{F,G} \leftarrow e^{-\frac{1}{K}} \times Z_{\emptyset, G_1} \times Z_{\emptyset, G_2}$$

• If $F = \begin{array}{c} \textcircled{2} \\ \text{---} \\ \boxed{F_1} \end{array} \boxed{F_2}$ and $G = \begin{array}{c} \textcircled{2} \\ \text{---} \\ \boxed{G_1} \end{array} \boxed{G_2}$, then

$$Z_{F,G} \leftarrow Z_{F_1, G_1} \times Z_{F_2, G_2} + \sum_{G = G' \circ G''} e^{-\frac{1}{K}} Z_{F_1, G'} Z_{F_2, G''} + \sum_{F = F' \circ F''} e^{-\frac{1}{K}} Z_{F'_1, G_1} Z_{F''_1, G_2}$$

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 $O(n_1 n_2)$.

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.

→ Simpler decomposition for tree alignments?

