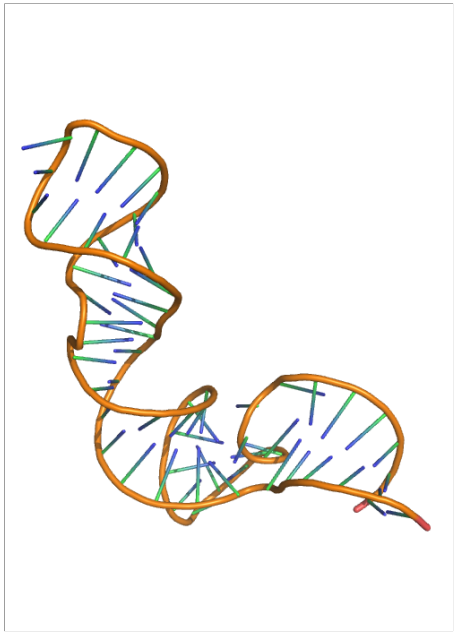
Extending the hypergraph analogy for RNA dynamic programming

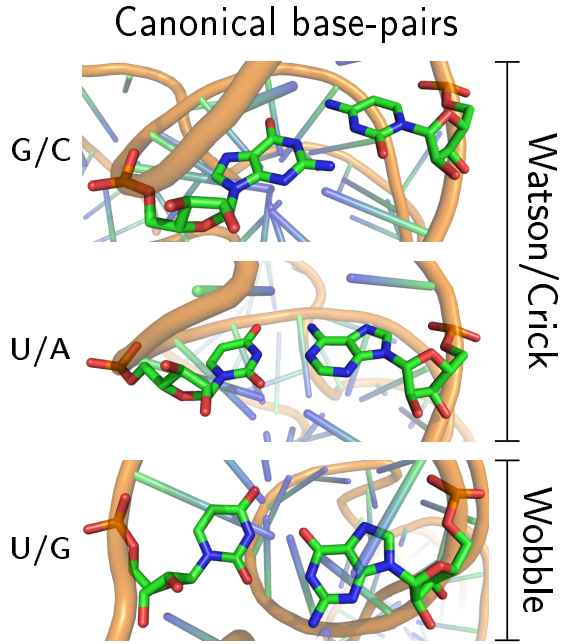
Yann Ponty Balaji Raman Cédric Saule

Polytechnique/CNRS/INRIA AMIB – France

RNA Folding

RNA = Biopolymer composed of nucleotides A, C, G, and U
A : Adenosine, C : Cytosine, G : Guanine et U : Uracil





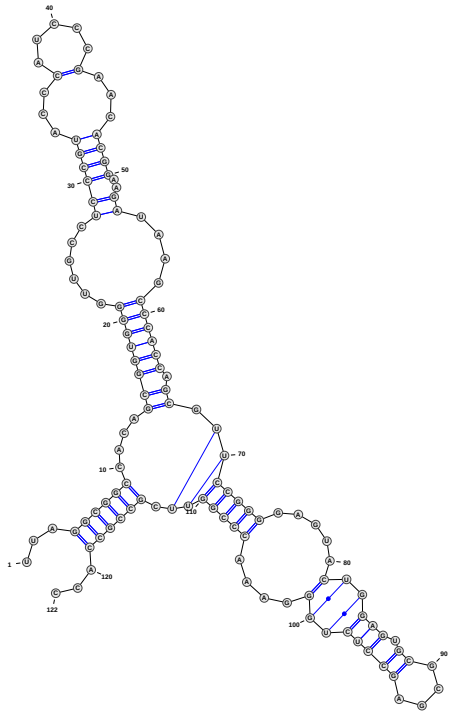
RNA folding = Stochastic continuous process directed by (resulting) the pairing of nucleotides.

Understanding RNA folding is a key step toward understanding and predicting its function(s).

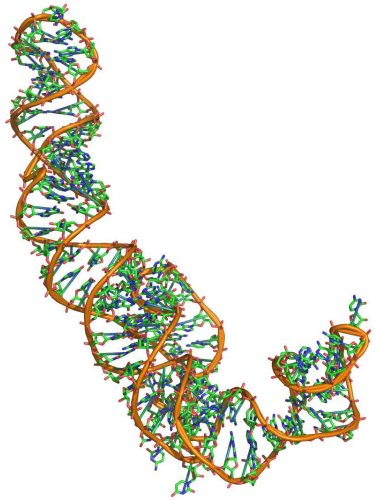
Three[1] levels of representation:

```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC
```

Primary structure



Secondary structure



Tertiary structures

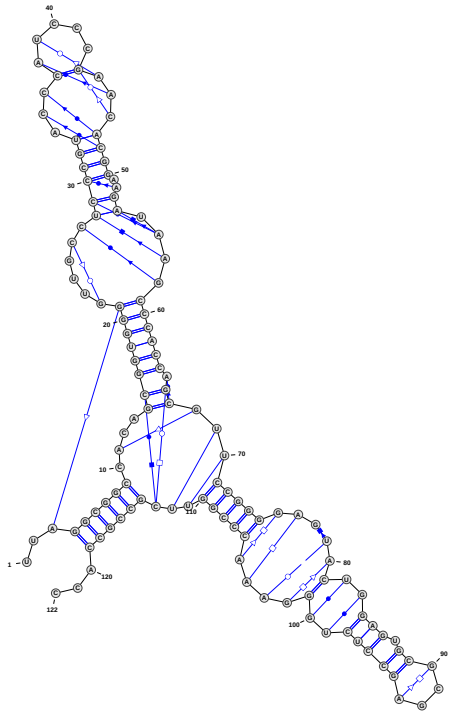
Source: 5s rRNA (PDB 1K73:B)

[1] Well, almost...

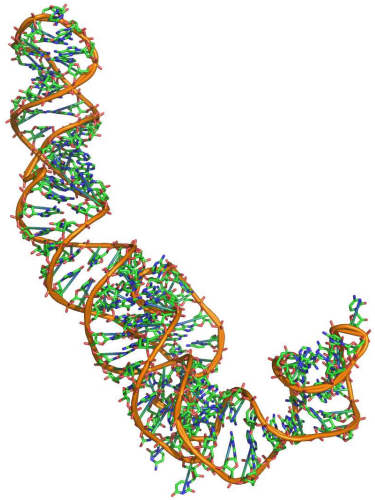
Three[1] levels of representation:

```
UUAGGCGGCCACAGC
GGUGGGGUUGCCUCC
CGUACCCAUCCCGAA
CACGGAAGAUAAGCC
CACCAGCGUUCCGGG
GAGUACUGGAGUGCG
CGAGCCUCUGGGAAA
CCCGGUUCGCCGCCA
CC
```

Primary structure



Secondary^{+} structure



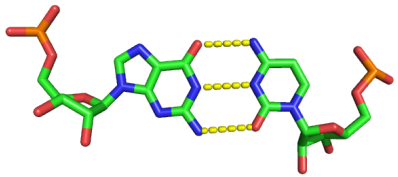
Tertiary structures

Source: 5s rRNA (PDB 1K73:B)

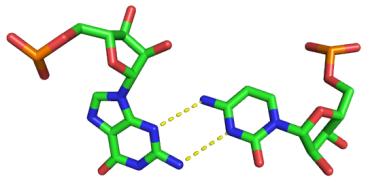
[1] Well, almost...

Discarded by secondary structure

- Non-canonical base-pairs
  Any basepair other than {(A-U), (C-G), (G-U)}
  Or interacting using a non-standard edge/orientation (WC/WC-Cis) [LW01].

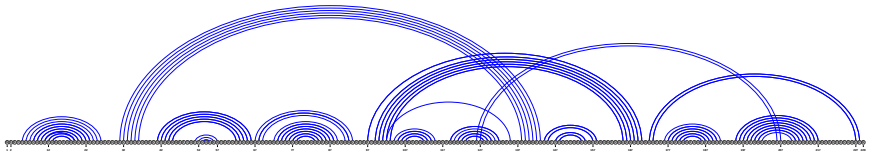


C/G canonical pair (WC/WC-Cis)



CG non-canonical pair (Sugar/WC-Trans)

- Pseudoknots



Pseudoknots within a group I Ribozyme (PDBID: 1Y0Q:A)

More expressive model, but *ab initio* folding with pseudoknots:
$\Rightarrow$ NP-Complete [LP00]... yet polynomial for restricted classes [CDR+04].

Our objective(s)

Authors	Complexity	Authors	Complexity
Lyngso and Pedersen	$\mathcal{O}(n^5)$	Cao and Chen	$\mathcal{O}(n^6)$
Reeder and Giegerich	$\mathcal{O}(n^4)$	Dirk and Pierce	$\mathcal{O}(n^5)$
Akutso and Uemara	$\mathcal{O}(n^5)$	Rivas and Eddy	$\mathcal{O}(n^6)$
Chen, Jabbari and Condon	$\mathcal{O}(n^5)$	...	...

Many DP algorithms proposed for folding with restricted pseudoknots.
Decompositions induce high complexities and/or are ambiguous, prohibiting ensemble-based approaches: Partition function, BP prob...

Is it possible to come up with highly expressive, unambiguous decompositions?

Our programme

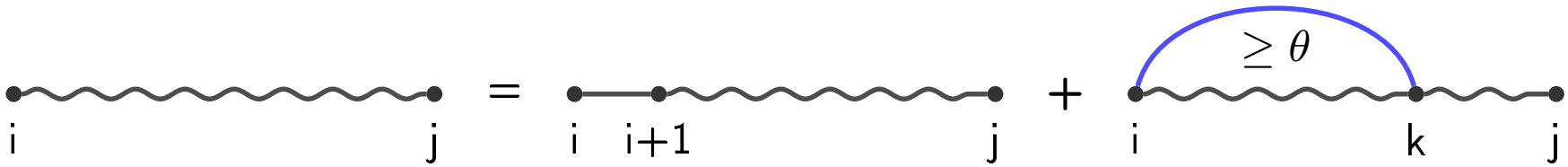
- To unify DP RNA algorithms into a single abstract framework.
- To reformulate ensemble-based applications within this framework.
- Improve existing algorithms by working on their decomposition at an abstract level.

May benefit to RNA-RNA interaction prediction, parameterized approaches...

Nearest-neighbor model

Free-energy = Weighted sum over base-pairs

Goal: Given RNA sequence ω, find Minimum Free Energy (MFE) secondary structure compatible with base-pairing.



$$\mathcal{N}_{i,t} = 0, \quad \forall t \in [i, i+\theta]$$

$$\mathcal{N}_{i,j} = \min \begin{cases} \mathcal{N}_{i+1,j} & i \text{ non apparié} \\ \min\limits_{k=i+\theta+1}^{j} E_{\omega_i,\omega_k} + \mathcal{N}_{i+1,k-1} + \mathcal{N}_{k+1,j} & i \text{ apparié à } k \end{cases}$$

$E_{G,C} = E_{C,G} = -3.0\,\frac{\text{kCal}}{\text{mol}}$, $E_{A,U} = E_{U,A} = -2.0\,\frac{\text{kCal}}{\text{mol}}$, and $E_{G,U} = E_{U,G} = -1.0\,\frac{\text{kCal}}{\text{mol}}$.

Once the minimal energy is figured out, the corresponding secondary structure is obtained through a backtrack procedure.

Turner model

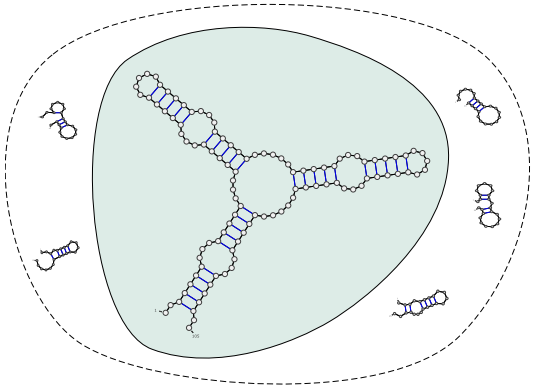
Free-energy = Weighted sum over loops

$$
\begin{aligned}
\mathcal{M}'_{i,j} &= \min \begin{cases} E_H(i,j) \\ E_S(i,j) + \mathcal{M}'_{i+1,j-1} \\ \mathrm{Min}_{i',j'}(E_{BI}(i,i',j',j) + \mathcal{M}'_{i',j'}) \\ a + c + \mathrm{Min}_k(\mathcal{M}_{i+1,k-1} + \mathcal{M}^1{}_{k,j-1}) \end{cases} \\
\mathcal{M}_{i,j} &= \mathrm{Min}_k \left\{ \min\left(\mathcal{M}_{i,k-1}, b(k-1)\right) + \mathcal{M}^1{}_{k,j} \right\} \\
\mathcal{M}^1{}_{i,j} &= \mathrm{Min}_k \left\{ b + \mathcal{M}^1{}_{i,j-1}, c + \mathcal{M}'_{i,j} \right\}
\end{aligned}
$$

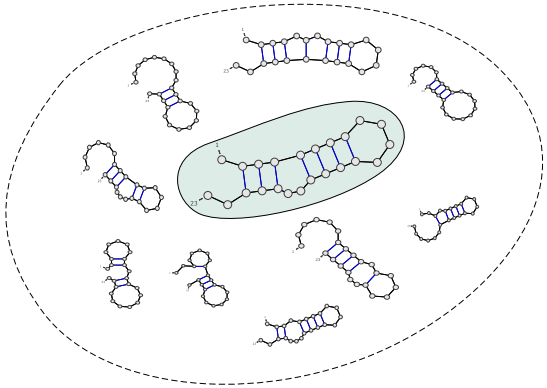
- $E_H(i,j)$: Energy of terminal loop *closed* by base-pair (i,j)
- $E_{BI}(i,j)$: Energy of bulge/internal loop renflement *closed* by bp (i,j)
- $E_S(i,j)$: Stacking energy $(i,j)/(i+1,j-1)$
- a,c,b: Penalties for multiple loops, helix and unpaired base in multiloop.

Beyond minimization

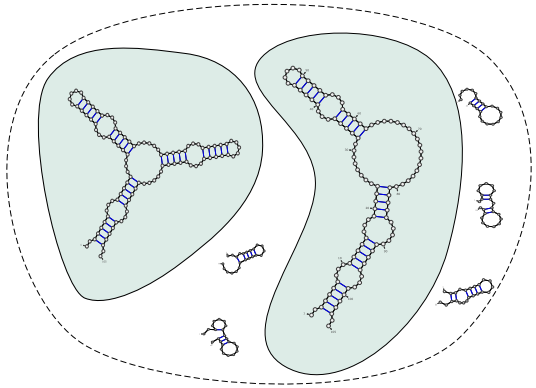
Additional motivation: Go beyond energy minimization.
Current driving hypothesis = Boltzmann ensemble of low energy



Functional folding?



Ill-defined folding:
mRNA?



Bistable RNA
Kinetics?

Decompositions will have to be unambiguous...

Dynamic programming

Starting from an instance, or *problem*:

- Search space: May depend on instance
- Objective function: Defined on search space, may depend on instance
- Dynamic programming equation: Relates value of objective function for problem to that of its sub-problems (substructure property), relying on a decomposition of search space

Compute obj. fun. for sub-problems (Use backtrack to build solution).

Nussinov example:

- Search space: Secondary structures inducing valid bps.
- Objective function: Free-energy
- DP equation/decomposition: MFE obtained by minimizing energy on subsequence(s)

Alternative view: DP equation generates search space AND implements a specific application.
$\Rightarrow$ Detach two conceptually different entities.

Parsing approaches

Search space modeled by context-free grammar(s) (CFG).

(Multitape)-attribute grammars (Lefebvre, Waldispühl *et al*):

Compute and minimize score based on attribute algebra.
Pros: Captures simple pseudoknots through *synchronized* multiple tapes.
Cons: Designed for optimization. Multi-tapes can be overkill for *almost* CFG.

Algebraic Dynamic Programming (Giegerich *et al*)

Pros: Adresses general applications thanks to an algebraic approach.
Cons: Pseudoknots require hacking the formalism.

Other

Hypergraphs (. . . , Roytberg/Finkelstein)

Pros: Very flexible. Mild influence of order. Any CFG can be transformed into equivalent hypergraph.
Cons: No generic implementation (yet!).

Parsing approaches

Search space modeled by context-free grammar(s) (CFG).

(Multitape)-attribute grammars (Lefebvre, Waldispühl *et al*):

Compute and minimize score based on attribute algebra.
Pros: Captures simple pseudoknots through *synchronized* multiple tapes.
Cons: Designed for optimization. Multi-tapes can be overkill for *almost* CFG.

Algebraic Dynamic Programming (Giegerich *et al*)

Pros: Adresses general applications thanks to an algebraic approach.
Cons: Pseudoknots require hacking the formalism.

Other

Hypergraphs (. . . , Roytberg/Finkelstein)

Pros: Very flexible. Mild influence of order. Any CFG can be transformed into equivalent hypergraph.
Cons: No generic implementation (yet!).

Parsing approaches

Search space modeled by context-free grammar(s) (CFG).

(Multitape)-attribute grammars (Lefebvre, Waldispühl *et al*):

Compute and minimize score based on attribute algebra.
Pros: Captures simple pseudoknots through *synchronized* multiple tapes.
Cons: Designed for optimization. Multi-tapes can be overkill for *almost* CFG.

Algebraic Dynamic Programming (Giegerich *et al*)

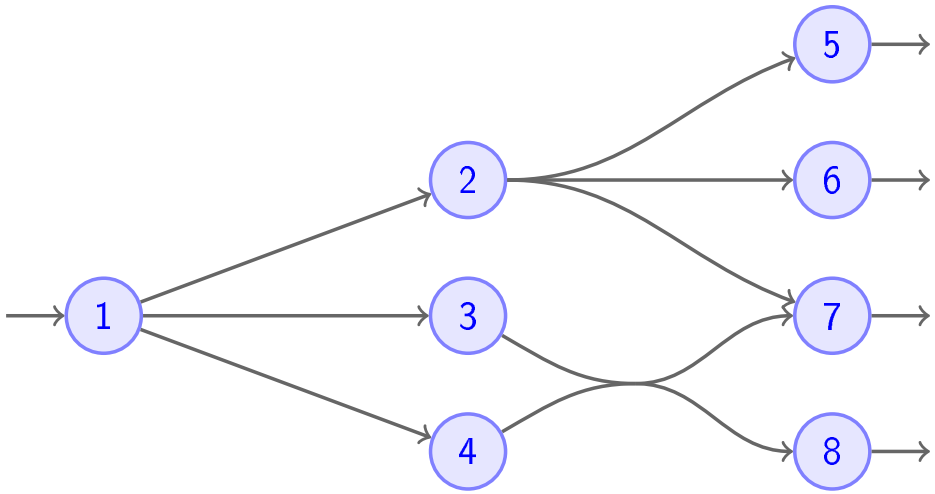
Pros: Adresses general applications thanks to an algebraic approach.
Cons: Pseudoknots require hacking the formalism.

Other

Hypergraphs (. . . , Roytberg/Finkelstein)

Pros: Very flexible. Mild influence of order. Any CFG can be transformed into equivalent hypergraph.
Cons: No generic implementation (yet!).

Hypergraphs



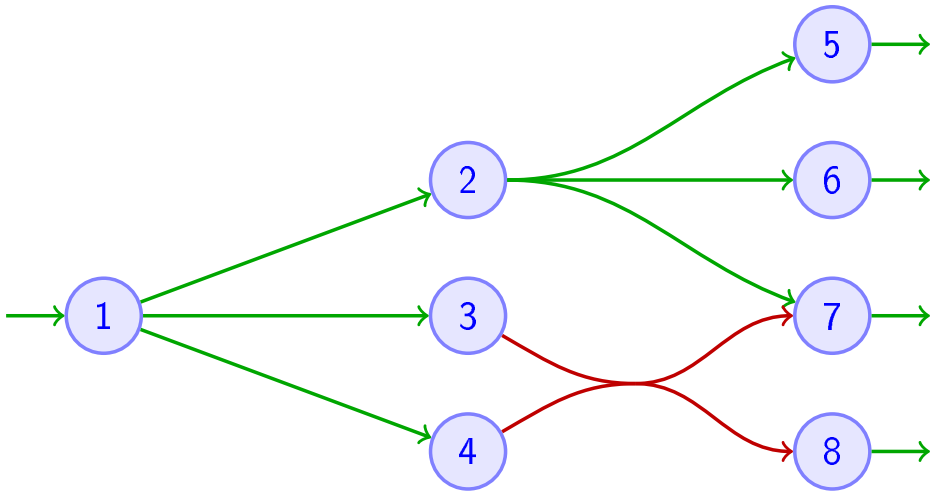
Hypergaphs generalize directed graphs to arcs of arbitrary in/out degrees.

Definition (Hypergraph)

A directed hypergaph $\mathcal{H}$ is a couple (V, E) such that:

- V is a set of vertices
- E is a set of hyperarcs $e = (t(e) \to h(e))$ such that $t(e), h(e) \subset E$

Forward hypergraphs, or F-graphs, are hypergraphs whose arcs have ingoing degree exactly 1.



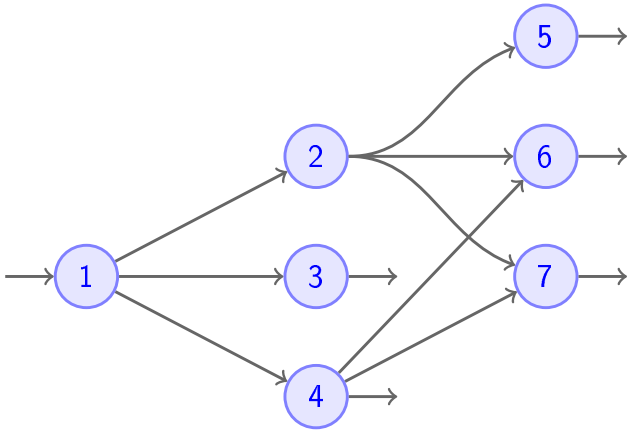
Hypergaphs generalize directed graphs to arcs of arbitrary in/out degrees.

Definition (Hypergraph)

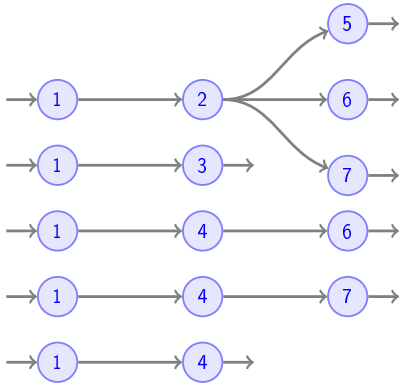
A directed hypergaph $\mathcal{H}$ is a couple (V, E) such that:

- V is a set of vertices
- E is a set of hyperarcs $e = (t(e) \to h(e))$ such that $t(e), h(e) \subset E$

Forward hypergraphs, or F-graphs, are hypergraphs whose arcs have ingoing degree exactly 1.



F-graph



All F-paths starting from vertex 1

Definition (F-path)

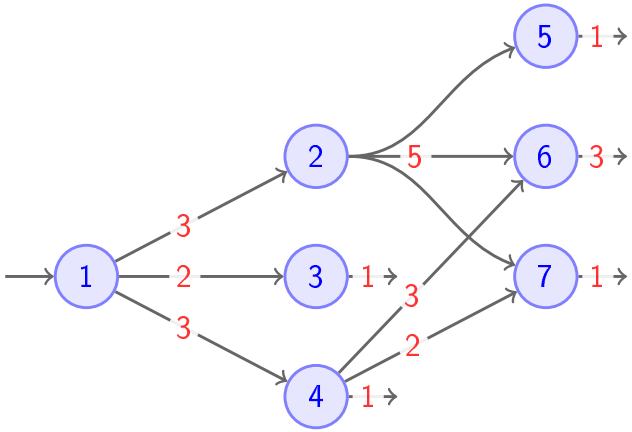
A F-path is a tree having root $s \in V$, whose children are F-paths built from the outgoing vertices of some arc $e = (s \to \mathbf{t}) \in E$.

Remark: Vertices of out degree 0 ($\mathbf{t} = \varnothing$) provide a terminal case to the above recursive definition.

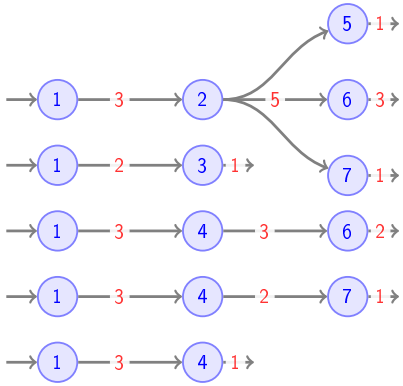
F-graph is independent iff any arc is present at most once in each F-path.

A numerical value fonction $\pi : E \to \mathbb{R}$ can be assigned to each arc $e \in E$:

- Weight of a path is the product of its arcs' values
- Score of a path is the sum of its arcs' values



F-graph



All F-paths starting from vertex 1

Definition (F-path)

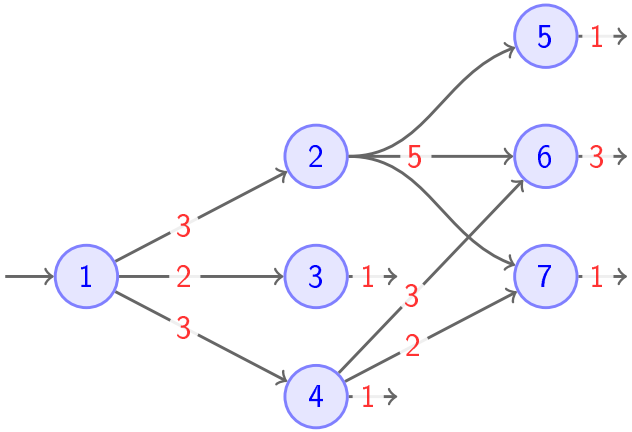
A F-path is a tree having root $s \in V$, whose children are F-paths built from the outgoing vertices of some arc $e = (s \rightarrow \mathbf{t}) \in E$.

Remark: Vertices of out degree 0 ($\mathbf{t} = \varnothing$) provide a terminal case to the above recursive definition.

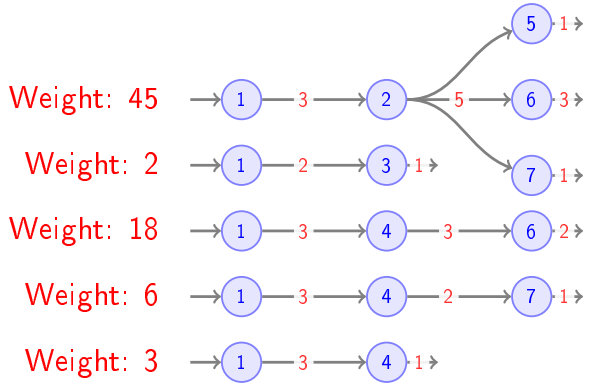
F-graph is independent iff any arc is present at most once in each F-path.

A numerical value fonction $\pi : E \rightarrow \mathbb{R}$ can be assigned to each arc $e \in E$:

- Weight of a path is the product of its arcs' values
- Score of a path is the sum of its arcs' values



F-graph



All F-paths starting from vertex 1

Definition (F-path)

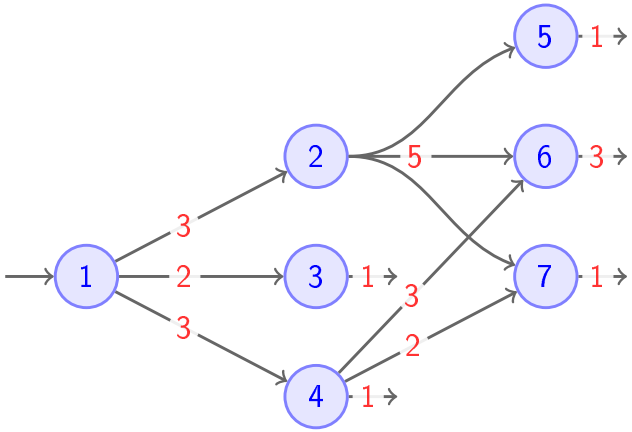
A F-path is a tree having root $s \in V$, whose children are F-paths built from the outgoing vertices of some arc $e = (s \rightarrow \mathbf{t}) \in E$.

Remark: Vertices of out degree 0 ($\mathbf{t} = \varnothing$) provide a terminal case to the above recursive definition.

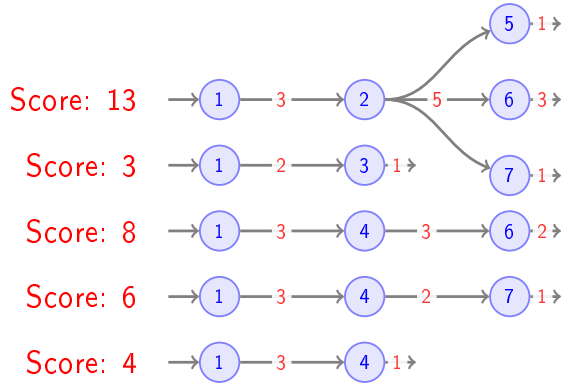
F-graph is independent iff any arc is present at most once in each F-path.

A numerical value fonction $\pi : E \rightarrow \mathbb{R}$ can be assigned to each arc $e \in E$:

- Weight of a path is the product of its arcs' values
- Score of a path is the sum of its arcs' values



F-graph



All F-paths starting from vertex 1

Definition (F-path)

A F-path is a tree having root $s \in V$, whose children are F-paths built from the outgoing vertices of some arc $e = (s \to \mathbf{t}) \in E$.

Remark: Vertices of out degree 0 ($\mathbf{t} = \varnothing$) provide a terminal case to the above recursive definition.

F-graph is independent iff any arc is present at most once in each F-path.

A numerical value fonction $\pi : E \to \mathbb{R}$ can be assigned to each arc $e \in E$:

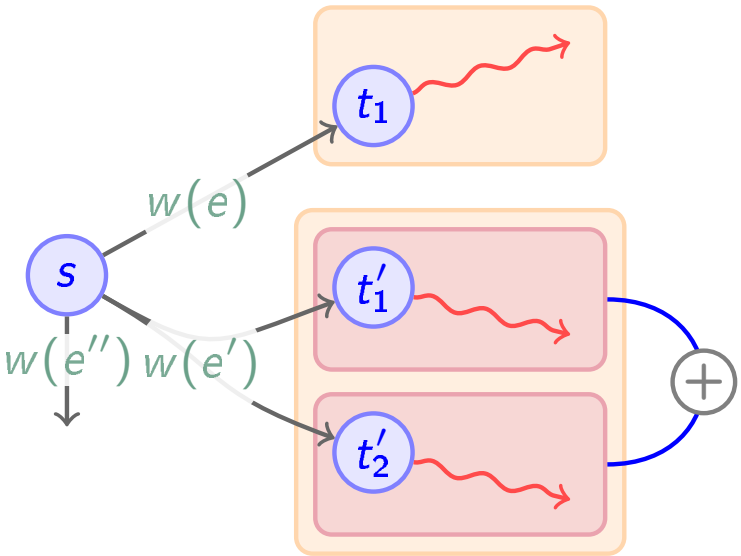
- Weight of a path is the product of its arcs' values
- Score of a path is the sum of its arcs' values

$\mathcal{H} = (s_0, V, E, \pi)$: acyclic hypergraph s_0: Initial node π: value function

Some questions naturally arise:

- What is the (min/max)imal score m_s of an F-path starting from $s \in V$?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory.
- What is the number n_s of F-paths starting from $s \in V$?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory.
- What is the total weight w_s of all F-paths starting from $s \in V$?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory.

$$m_s = \min_{e=(s \to \mathbf{t}) \in E} \left(w(e) + \sum_{u \in \mathbf{t}} m_u \right)$$

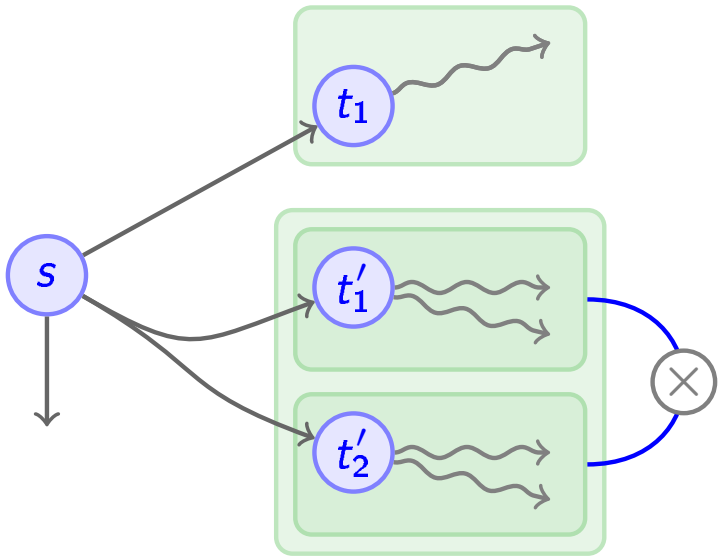


$\mathcal{H} = (s_0, V, E, \pi)$: acyclic hypergraph s_0: Initial node π: value function

Some questions naturally arise:

- What is the (min/max)imal score m_s of an F-path starting from $s \in V$?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory.
- What is the number n_s of F-paths starting from $s \in V$?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory.
- What is the total weight w_s of all F-paths starting from $s \in V$?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory.

$$n_s = \sum_{(s \to \mathbf{t}) \in E} \prod_{u \in \mathbf{t}} n_u$$

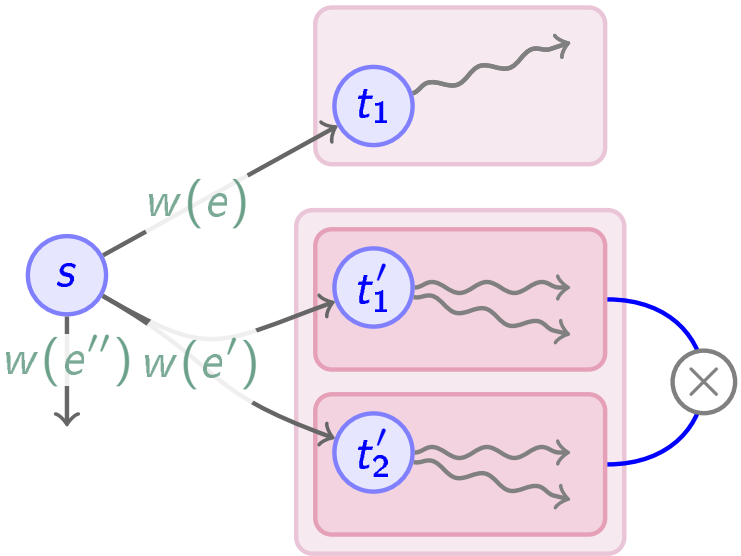


$\mathcal{H} = (s_0, V, E, \pi)$: acyclic hypergraph s_0: Initial node π: value function

Some questions naturally arise:

- What is the (min/max)imal score m_s of an F-path starting from $s \in V$?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory.
- What is the number n_s of F-paths starting from $s \in V$?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory.
- What is the total weight w_s of all F-paths starting from $s \in V$?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory.

$$w_s = \sum_{e=(s \to \mathbf{h}(e)) \in E} w(e) \cdot \prod_{s' \in \mathbf{h}(e)} w_{s'}$$



Definition (Weighted distribution)

Assume a weighted, *Boltzmann-like*, distribution on the set $\mathcal{T}$ of F-Paths:

$$\mathbb{P}(p|\pi) = \frac{\prod_{e \in p} \pi(e)}{w_{s_0}}, \forall p \in \mathcal{T}.$$

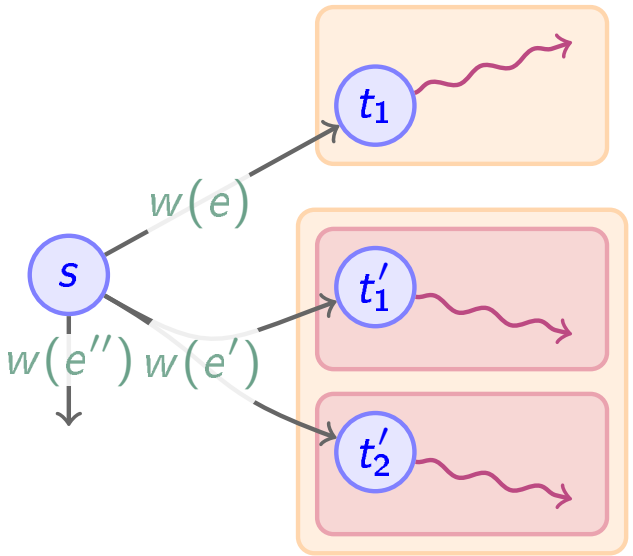
Ensemble related – questions arise:

- How to generate a random F-path p from $\mathcal{T}$ w.r.t. weighted distribution?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory precomputation + $\mathcal{O}(|p| + \sum_{e \in p} |\mathbf{h}(e)|)$ time generation.
- What is the probability that a given arc be present in a random F-Path?
  $\Rightarrow$ Inside/outside algorithm.
- Distribution of additive features

Algorithm:
Choose $e_i = (s \rightarrow \mathbf{t}_i)$ with probability $p_{s,i}$ such that:

$$p_{s,i} = \frac{w(e) \cdot \prod_{s' \in \mathbf{t}} w_{s'}}{w_s}$$

Iterate the process over all $\mathbf{t}_i$'s.



Definition (Weighted distribution)

Assume a weighted, *Boltzmann-like*, distribution on the set $\mathcal{T}$ of F-Paths:

$$\mathbb{P}(p|\pi) = \frac{\prod_{e \in p} \pi(e)}{W_{s_0}}, \forall p \in \mathcal{T}.$$

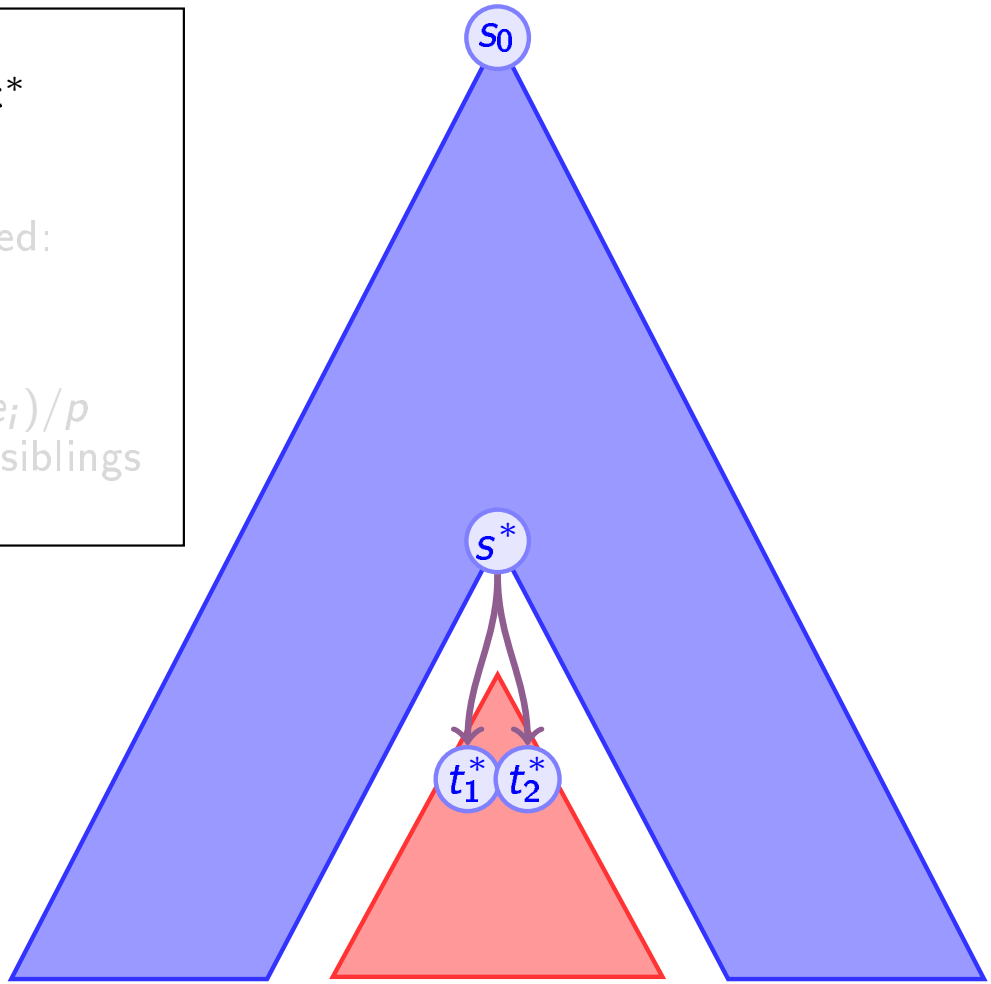
Ensemble related – questions arise:

- How to generate a random F-path p from $\mathcal{T}$ w.r.t. weighted distribution?
  $\Rightarrow$ Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory precomputation + $\mathcal{O}(|p| + \sum_{e \in p} |\mathbf{h}(e)|)$ time generation.
- What is the probability that a given arc be present in a random F-Path?
  $\Rightarrow$ Inside/outside algorithm.
- Distribution of additive features

Inside/outside algorithm

F-path using a given arc $e^*(s^* \rightarrow \mathbf{t}^*)$

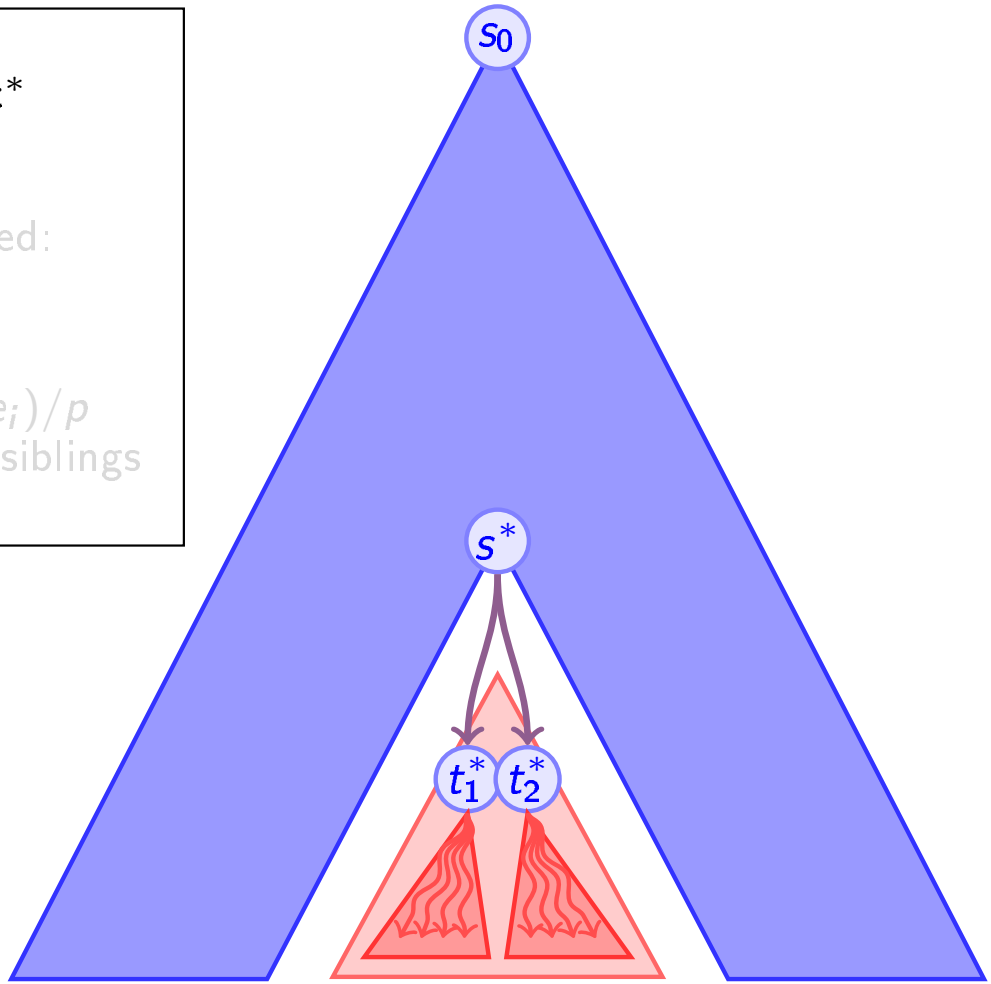
- Inside: Some F-path built from $\mathbf{t}^*$
- Distinguished arc e^*
- Outside: Context where e^* is used:
  - Path $p = (s_0, s_1, \ldots, s^*)$ using F-arcs $(e_1, \ldots, e_k)$
  - For each vertex in some $h(e_i)/p$ complete F-path by adding siblings to the vertices in p.



Inside/outside algorithm

F-path using a given arc $e^*(s^* \rightarrow \mathbf{t}^*)$

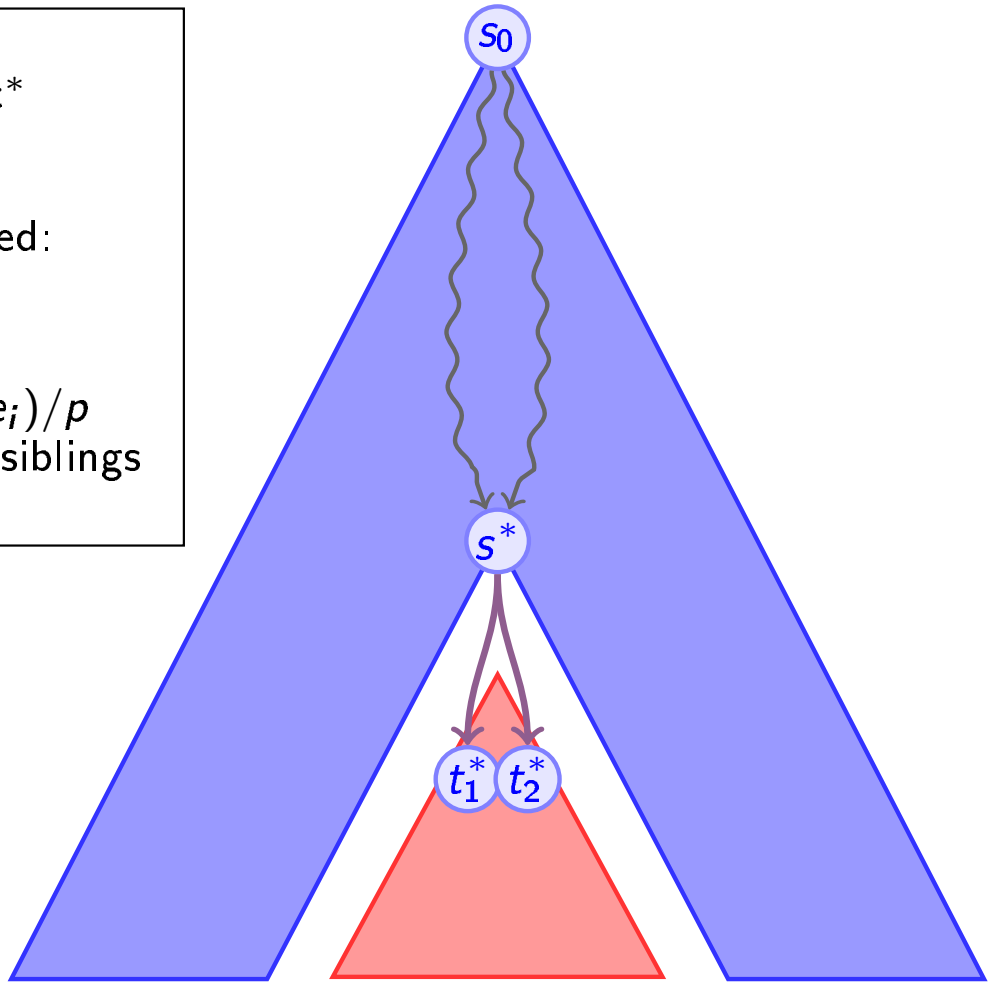
- Inside: Some F-path built from $\mathbf{t}^*$
- Distinguished arc e^*
- Outside: Context where e^* is used:
  - Path $p = (s_0, s_1, \ldots, s^*)$ using F-arcs $(e_1, \ldots, e_k)$
  - For each vertex in some $h(e_i)/p$ complete F-path by adding siblings to the vertices in p.



Inside/outside algorithm

F-path using a given arc $e^*(s^* \rightarrow \mathbf{t}^*)$

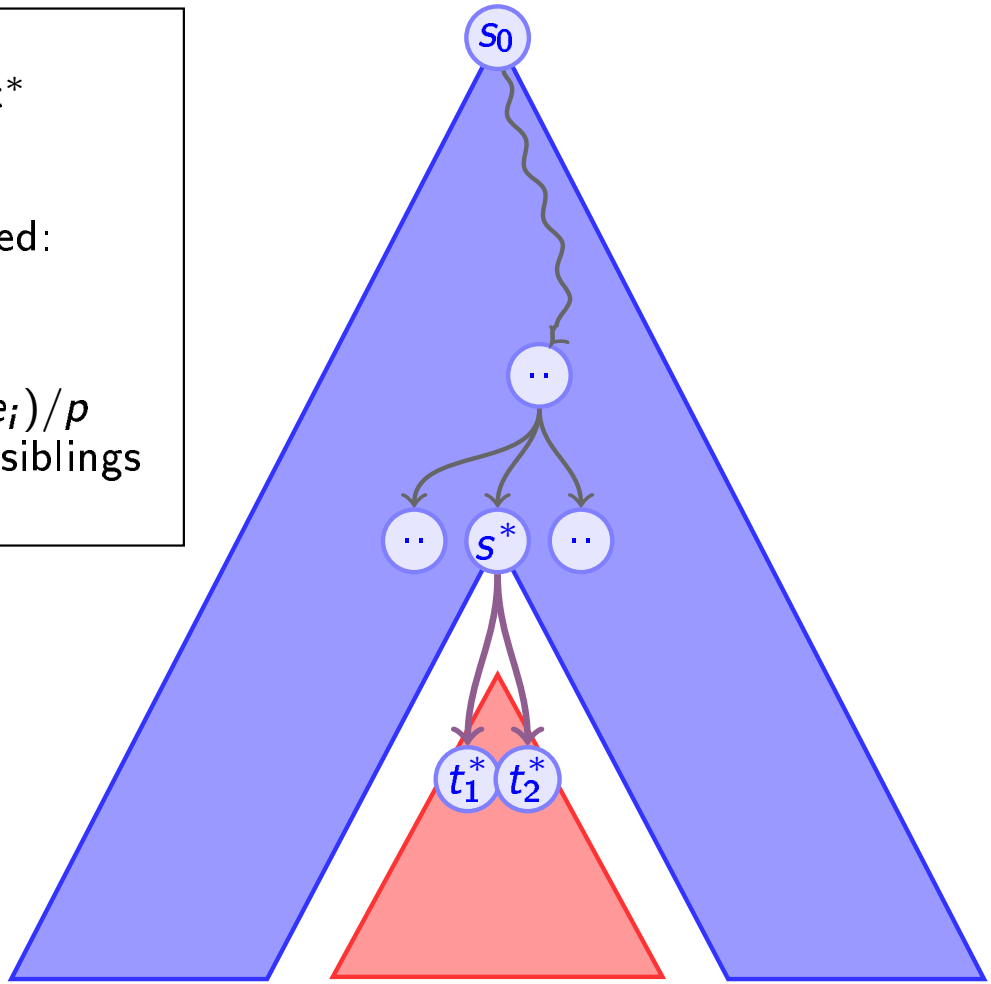
- Inside: Some F-path built from $\mathbf{t}^*$
- Distinguished arc e^*
- Outside: Context where e^* is used:
  - Path $p = (s_0, s_1, \ldots, s^*)$ using F-arcs $(e_1, \ldots, e_k)$
  - For each vertex in some $h(e_i)/p$ complete F-path by adding siblings to the vertices in p.



Inside/outside algorithm

F-path using a given arc $e^*(s^* \rightarrow \mathbf{t}^*)$

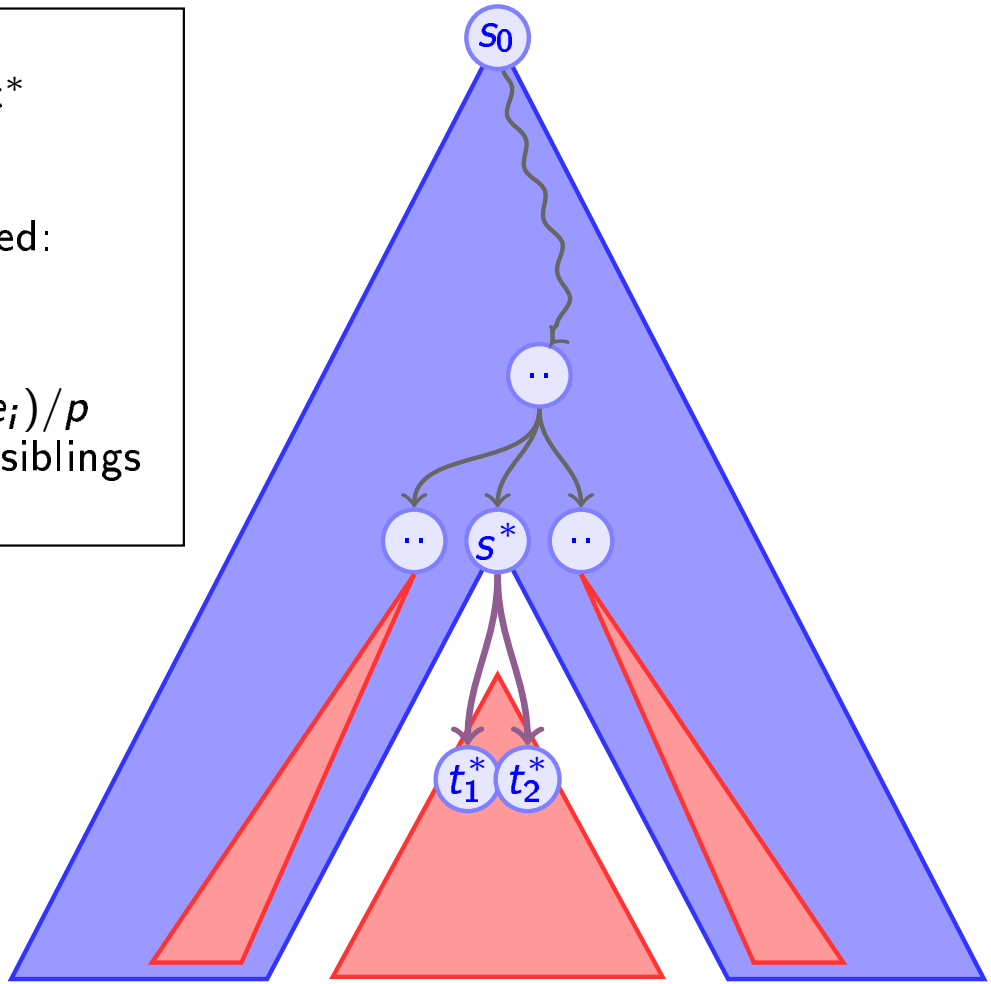
- Inside: Some F-path built from $\mathbf{t}^*$
- Distinguished arc e^*
- Outside: Context where e^* is used:
  - Path $p = (s_0, s_1, \ldots, s^*)$ using F-arcs $(e_1, \ldots, e_k)$
  - For each vertex in some $h(e_i)/p$ complete F-path by adding siblings to the vertices in p.



Inside/outside algorithm

F-path using a given arc $e^*(s^* \rightarrow \mathbf{t}^*)$

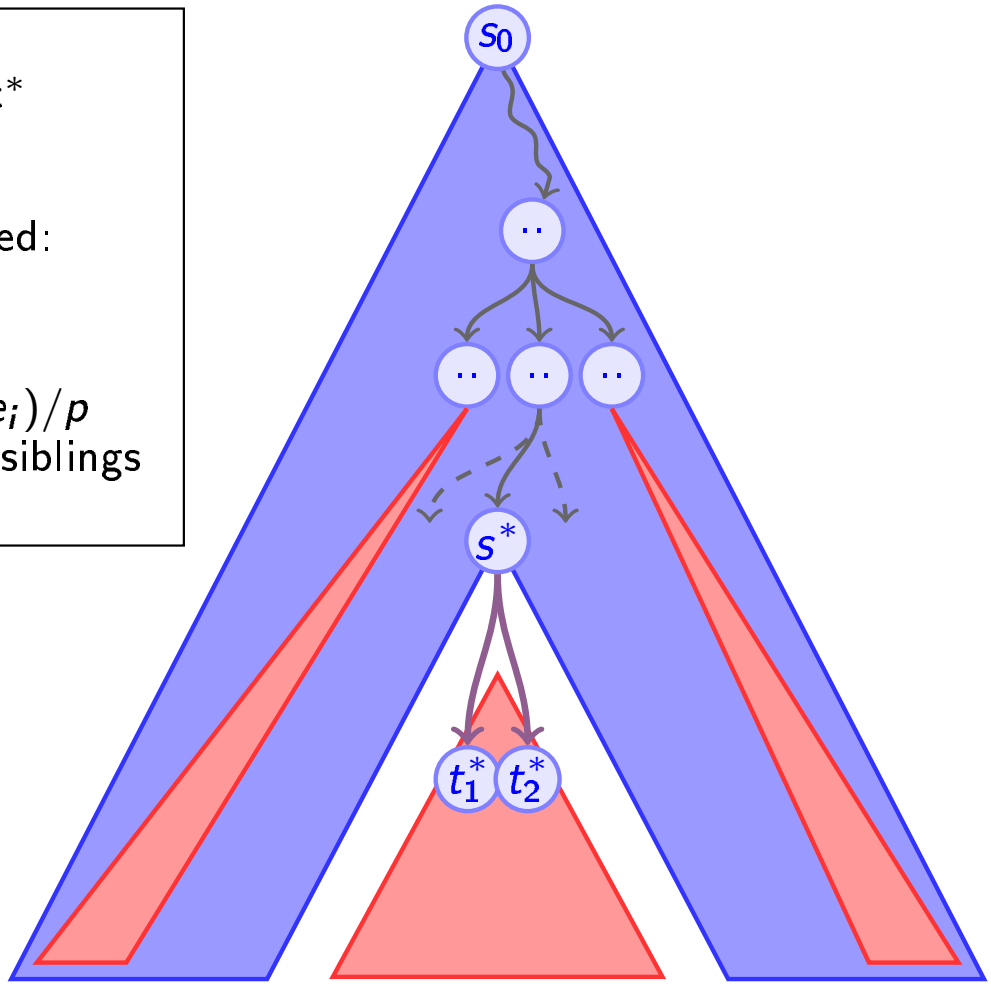
- Inside: Some F-path built from $\mathbf{t}^*$
- Distinguished arc e^*
- Outside: Context where e^* is used:
  - Path $p = (s_0, s_1, \ldots, s^*)$ using F-arcs $(e_1, \ldots, e_k)$
  - For each vertex in some $h(e_i)/p$ complete F-path by adding siblings to the vertices in p.



Inside/outside algorithm

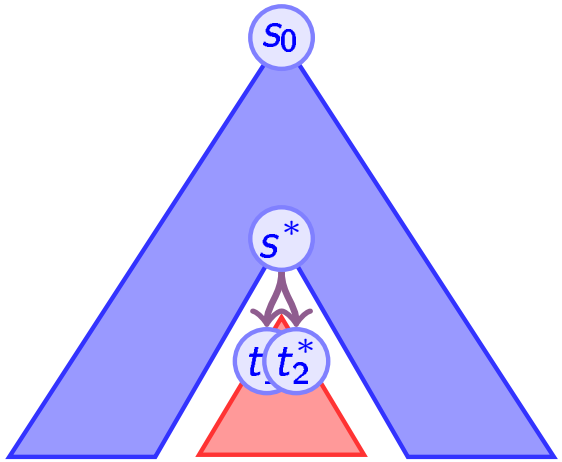
F-path using a given arc $e^*(s^* \rightarrow \mathbf{t}^*)$

- Inside: Some F-path built from $\mathbf{t}^*$
- Distinguished arc e^*
- Outside: Context where e^* is used:
  - Path $p = (s_0, s_1, \ldots, s^*)$ using F-arcs $(e_1, \ldots, e_k)$
  - For each vertex in some $h(e_i)/p$ complete F-path by adding siblings to the vertices in p.



Inside/outside algorithm

F-path using a given arc $e^*(s^* \to \mathbf{t}^*)$
- Inside: Some F-path built from $\mathbf{t}^*$
- Distinguished arc e^*
- Outside: Context where e^* is used:
  - Path $p = (s_0, s_1, \ldots, s^*)$ using F-arcs $(e_1, \ldots, e_k)$
  - For each vertex in some $h(e_i)/p$ complete F-path by adding siblings to the vertices in p.



If F-graph is acyclic and independent, this decomposition is complete and unambiguous, and induces the following DP equations for the cumulated probability p_{e^*} of all F-paths featuring $e^* = (s^* \to \mathbf{t}^*)$:

$$p_{e^*} = \frac{b_{s^*} \cdot \pi(e) \cdot \prod_{s' \in t^*} w_{s'}}{w_{s_0}}$$

$$b_s = \mathbf{1}_{s=s_0} + \sum_{\substack{e'=(s' \to \mathbf{t}') \in E \\ \text{s. t. } s \in \mathbf{t}}} \pi(e') \cdot b_{s'} \cdot \prod_{\substack{s'' \in \mathbf{t}' \\ s'' \neq s}} w_{s''}, \quad \forall s \in V$$

Definition (Weighted distribution)

Assume a weighted, *Boltzmann-like*, distribution on the set $\mathcal{T}$ of F-Paths:

$$\mathbb{P}(p|\pi) = \frac{\prod_{e \in p} \pi(e)}{w_{s_0}}, \forall p \in \mathcal{T}.$$

Ensemble related – questions arise:

- How to generate a random F-path p from $\mathcal{T}$ w.r.t. weighted distribution?
  Complexities: $\Theta(|E| + |V|)$ time/$\Theta(|V|)$ memory precomputation
  $+ \mathcal{O}(|p| + \sum_{e \in p} |\mathbf{h}(e)|)$ time generation.
- What is the probability that a given arc be present in a random F-Path?
  $\Rightarrow$ Inside/outside algorithm
  Complexities: $\mathcal{O}(|V| + |E| + \sum_{e \in E} |\mathbf{h}(e)|^2)$time/$\Theta(|V|)$ memory
- How to characterize the distribution of some additive features?
  $\Rightarrow$ Extraction of generalized moments

Let $\mathcal{T}$ be the set of F-paths associated with a given hypergraph.

Definition (Feature)

A feature is a function $\alpha : E \to \mathbb{R}$ inherited additively by F-paths through

$$\alpha(p) = \sum_{e \in p} \alpha(e)$$

Example: Let us consider additive price α function over F-arcs and its associated random variable X_α.

A Within the weighted distribution, what is the expected price of an F-path?

B ...the variance of the price X_α of an F-path?

C How does the price X_α correlates with the weight X_π?

$$\text{A} = \frac{\sum_{p \in \mathcal{T}} \pi(p) \cdot \alpha(p)}{w_{s_0}} = \mathbb{E}(X_\alpha) \qquad \text{B} = \frac{\sum_{p \in \mathcal{T}} \pi(p) \cdot \alpha(p)^2}{w_{s_0}} - \mathbb{E}(X_\alpha)^2 = \mathbb{E}(X_\alpha^2) - \mathbb{E}(X_\alpha)^2$$

$$\text{C} = \frac{\mathbb{E}(X_\alpha \cdot X_\pi) - \mathbb{E}(X_\alpha)\mathbb{E}(X_\pi)}{\sqrt{(\mathbb{E}(X_\alpha^2) - \mathbb{E}(X_\alpha)^2)(\mathbb{E}(X_\pi^2) - \mathbb{E}(X_\pi)^2)}}$$

$\Rightarrow$ Can be expressed in term of the moments of the feature(s) distribution.
How to extract them?

Goal: To extract the (combined) moment of an additive feature :

$$\mathbb{E}(X_{\alpha_1}^{k_1} \cdots X_{\alpha_m}^{k_m}) = \sum_{p \in \mathcal{T}} \frac{\pi(p)}{w_{s_0}} \cdot \prod_{i=1}^{m} \alpha_i(p)^{k_m}$$

Difficulty: Mixing additive (features) and multiplicative algebraic aspects (weighted distribution).
Solution: Modify hypergraph to introduce controlled ambiguity.

Definition (Feature-pointed F-graph)

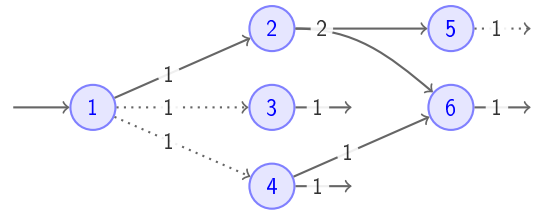
Let $\mathcal{H} = (s_0, V, E, \pi)$ be a weighted F-graph and $\alpha : E \to \mathbb{R}$ some feature. The α-pointing of $\mathcal{H}$ is an F-graph $\mathcal{H}^{\bullet\alpha} = (s_0^{\bullet\alpha}, V^{\bullet\alpha}, E^{\bullet\alpha}, \pi^{\bullet\alpha})$ such that:

- Pointed versions of vertices are introduced $\Rightarrow V^{\bullet\alpha} := V \cup \{s^{\bullet\alpha} \mid s \in V\}$
- New arcs are introduced to propagate a point or erase it.
  - Propagation $P^{\bullet\alpha} := \bigcup_{\substack{(s \to \mathbf{t}) \in E \\ \mathbf{t} \neq \varnothing}} \{s^{\bullet\alpha} \to (t_1, \ldots, t_i^{\bullet\alpha}, \ldots, t_k) \mid i \in [1, k]\}$
  - Point erasure $M^{\bullet\alpha} := \bigcup_{(s \to \mathbf{t}) \in E} \{(s^{\bullet\alpha} \to \mathbf{t})\}$

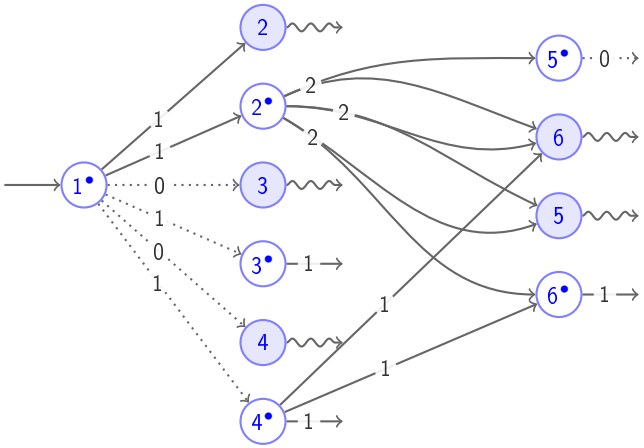
  $\Rightarrow E^{\bullet\alpha} := E \cup P^{\bullet\alpha} \cup M^{\bullet\alpha}$
- Weight function is extended to partially incorporate feature function

$$\pi^{\bullet\alpha}(e') := \begin{cases} \pi(e'^{\circ\alpha}) & \text{If } e' \in P \cup E \\ \alpha(e'^{\circ}) \cdot \pi(e'^{\circ\alpha}) & \text{Otherwise } (e' \in M^{\bullet\alpha}) \end{cases}, \quad \forall e' \in E^{\bullet\alpha}$$

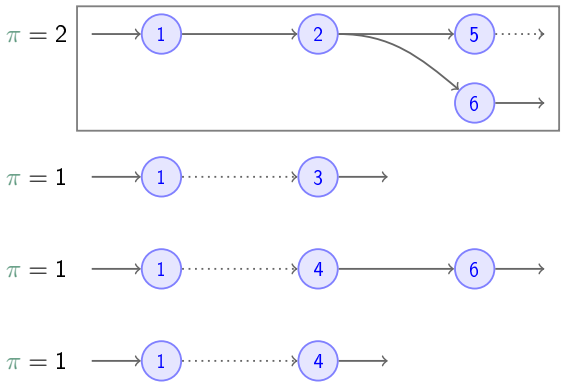
Example: Simple 0 (dashed) or 1 (plain) feature function α.



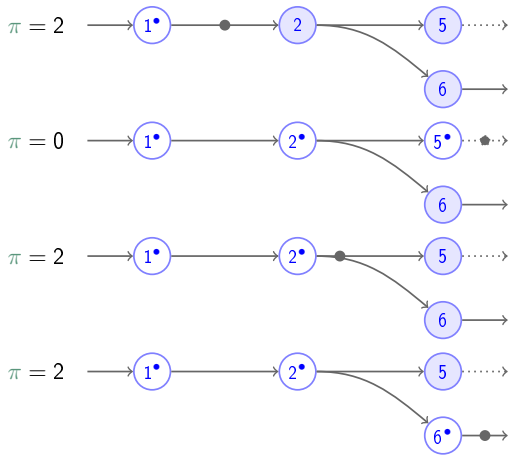
Initial F-graph



Pointed F-graph



Initial F-Paths



Analogs in $\mathcal{H}^\bullet$ of boxed F-path

Rationale: Through pointing, each F-path p in $\mathcal{H}$ is duplicated $|p|$ times in $\mathcal{H}^\bullet$ ($\Rightarrow$ analogs of p) . Each copy features a point erasure over a different F-arc.
$\Rightarrow$ The total weight, under $\pi^{\bullet\alpha}$, of all analogs of p is now $\pi(p) \cdot \alpha(p)$.

Extracting moments

Input: F-graph $\mathcal{H} = (s_0, V, E, \pi)$
Goal: To extract the (higher order) moment of an additive feature :

$$\mathbb{E}(X_{\alpha_1}^{k_1} \cdots X_{\alpha_m}^{k_m}) = \sum_{p \in \mathcal{T}} \frac{\pi(p)}{w_{s_0}} \cdot \prod_{i=1}^{m} \alpha_i(p)^{k_m}$$

Algorithm

- Apply weighted count algorithm to $\mathcal{H} \rightarrow w_{s_0}$.
- Point $\mathcal{H}$ repeatedly (commutative transform) $\rightarrow \mathcal{H}^\bullet$.
- Apply weighted count algorithm to $\mathcal{H}^\bullet$

$$\Rightarrow w^\bullet = \sum_{p \in \mathcal{T}} \pi(p) \cdot \prod_{i=1}^{m} \alpha_i(p)^{k_m}.$$

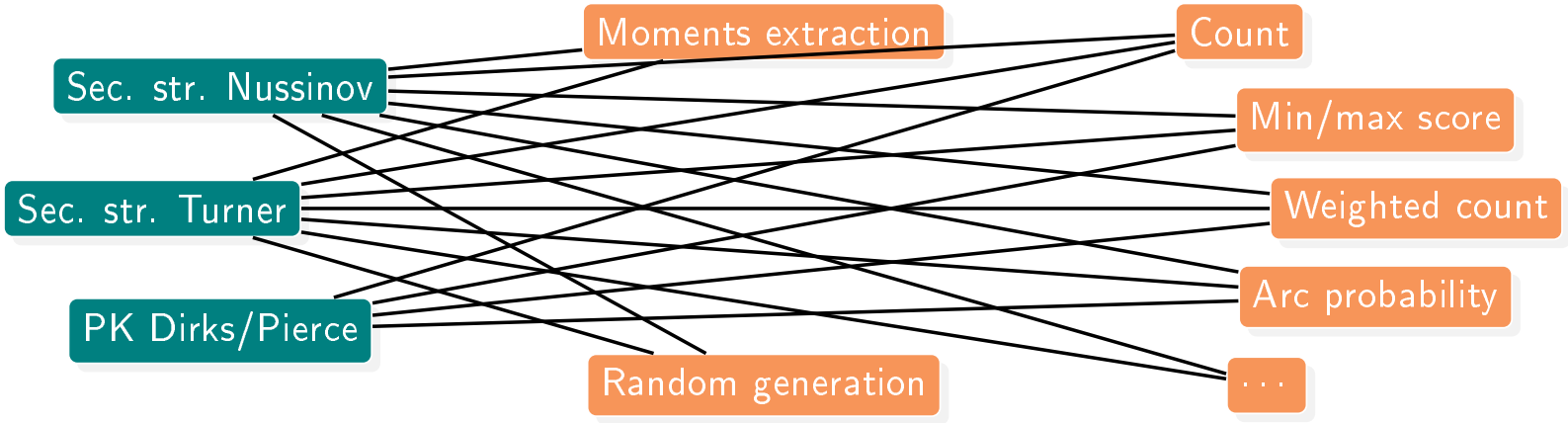
- Return $w^\bullet / w_{s_0}$.

Complexities: $\mathcal{O}\left(2^k \cdot |V| + \sum_{e \in E} (|\mathbf{h}(e)| + 1) \cdot (|\mathbf{h}(e)| + 2)^k\right)$ time
$\mathcal{O}(2^k \cdot |V|)$ memory, with $k = \sum_{i=1}^{m} k_i$.

Remarks: Can be improved by pointing k_i times in one go instead of pointing repeatedly. Without loss of generality, $|\mathbf{h}(e)| = 2$ can be assumed (CNF).

Message #1

Specific applications of Dynamic Programming could (and should) be detached from the equation, and be expressed at an abstract level.



Message #2

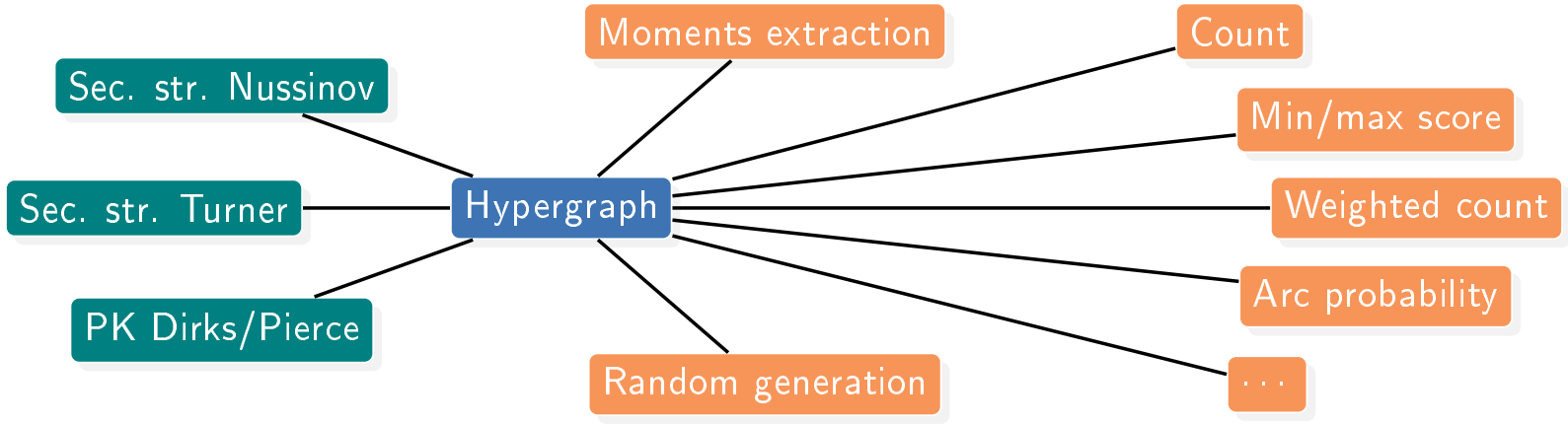
Thanks to a pointing operator (formal derivative), one can extract arbitrary moments of features distribution under a weighted/Boltzmann distribution.

Credits: Roytberg and Finkelstein for Hypergraph DP in Bioinformatics, L. Hwang for algebraic hypergraph DP, Flajolet *et al* for formal derivative through pointing...

Let us now address the construction of suitable F-graphs...

Message #1

Specific applications of Dynamic Programming could (and should) be detached from the equation, and be expressed at an abstract level.



Message #2

Thanks to a pointing operator (formal derivative), one can extract arbitrary moments of features distribution under a weighted/Boltzmann distribution.

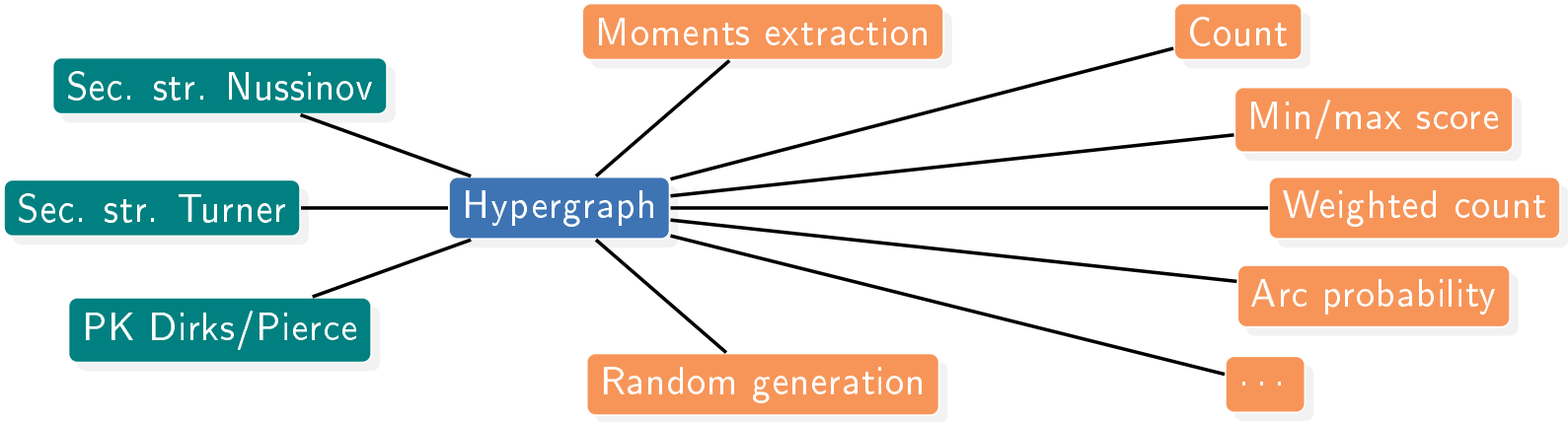
Credits: Roytberg and Finkelstein for Hypergraph DP in Bioinformatics, L. Hwang for algebraic hypergraph DP, Flajolet *et al* for formal derivative through pointing...

Let us now address the construction of suitable F-graphs...

Half time summary

Message #1

Specific applications of Dynamic Programming could (and should) be detached from the equation, and be expressed at an abstract level.



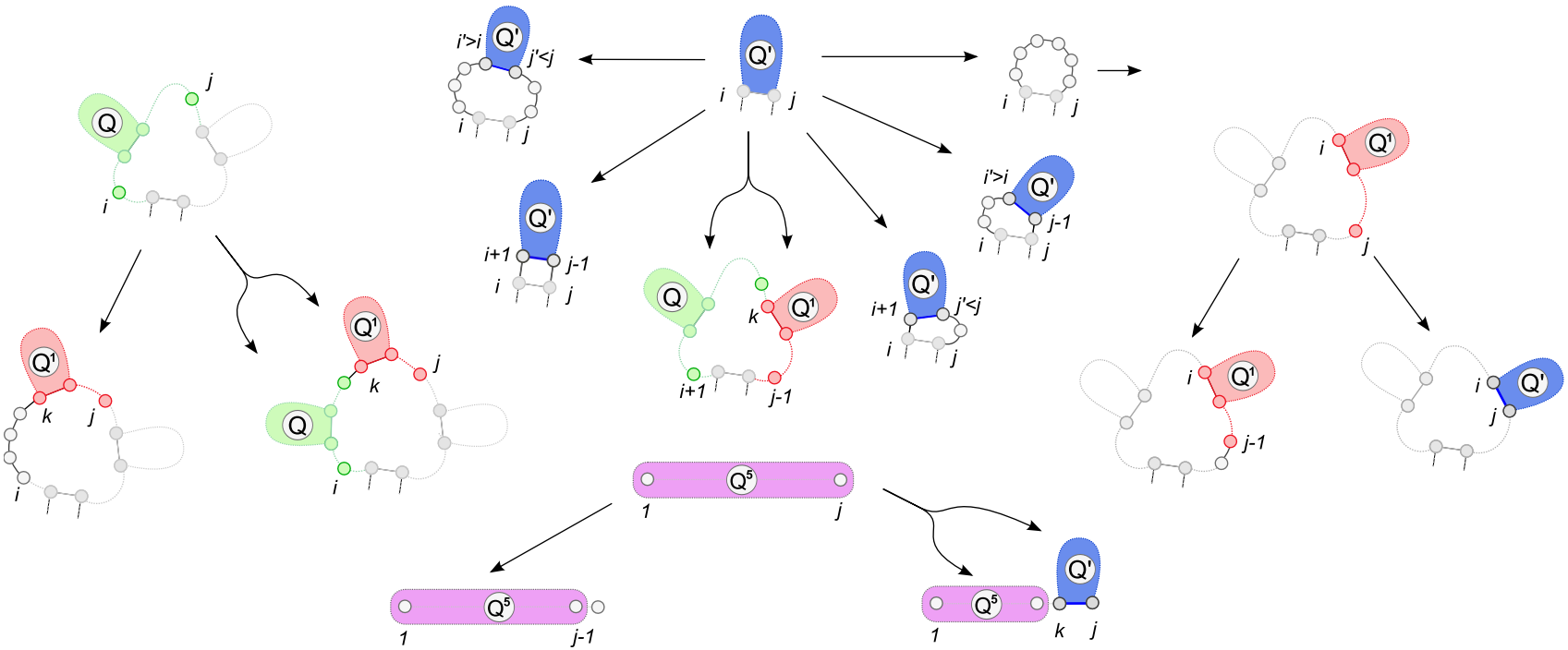
Message #2

Thanks to a pointing operator (formal derivative), one can extract arbitrary moments of features distribution under a weighted/Boltzmann distribution.

Credits: Roytberg and Finkelstein for Hypergraph DP in Bioinformatics, L. Hwang for algebraic hypergraph DP, Flajolet *et al* for formal derivative through pointing. . .

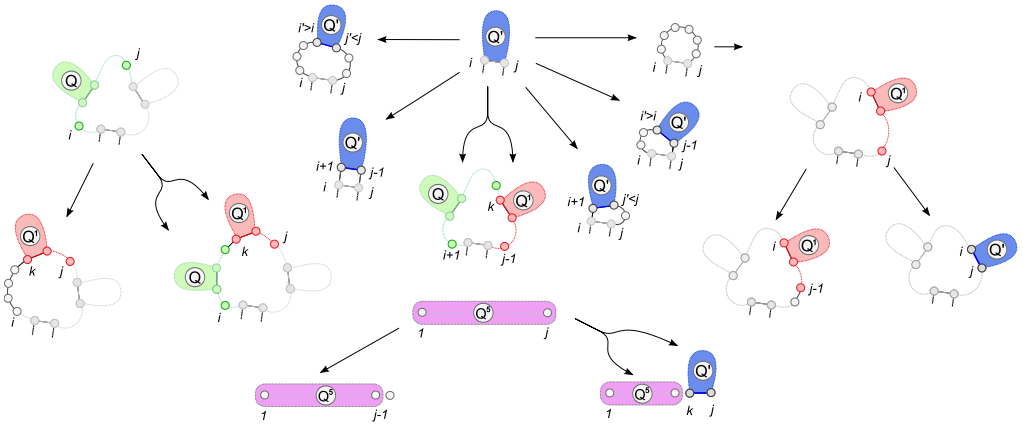
Let us now address the construction of suitable F-graphs. . .

Mfold/Unafold decomposition



This decomposition is provably unambiguous and complete, and yields an F-graph that is acyclic, independent, and has $\Theta(n^2)$ vertices and $\mathcal{O}(n^3)$ arcs.

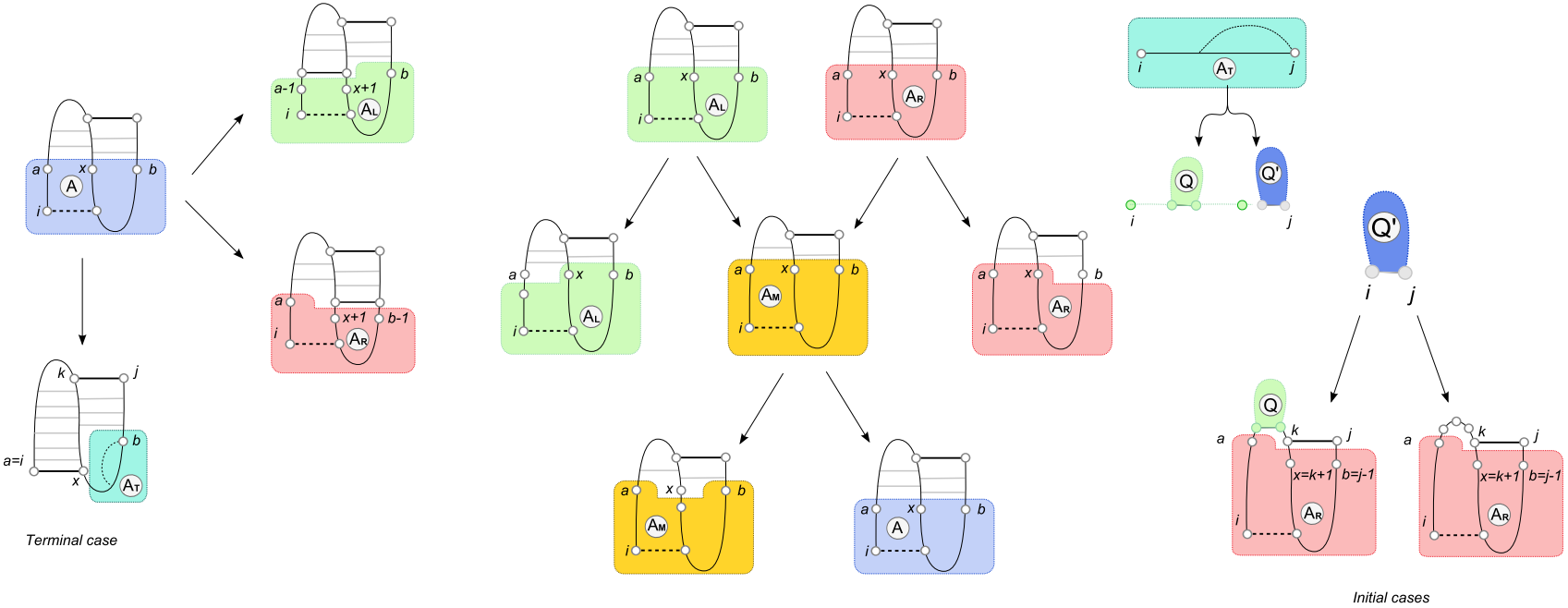
Mfold/Unafold decomposition



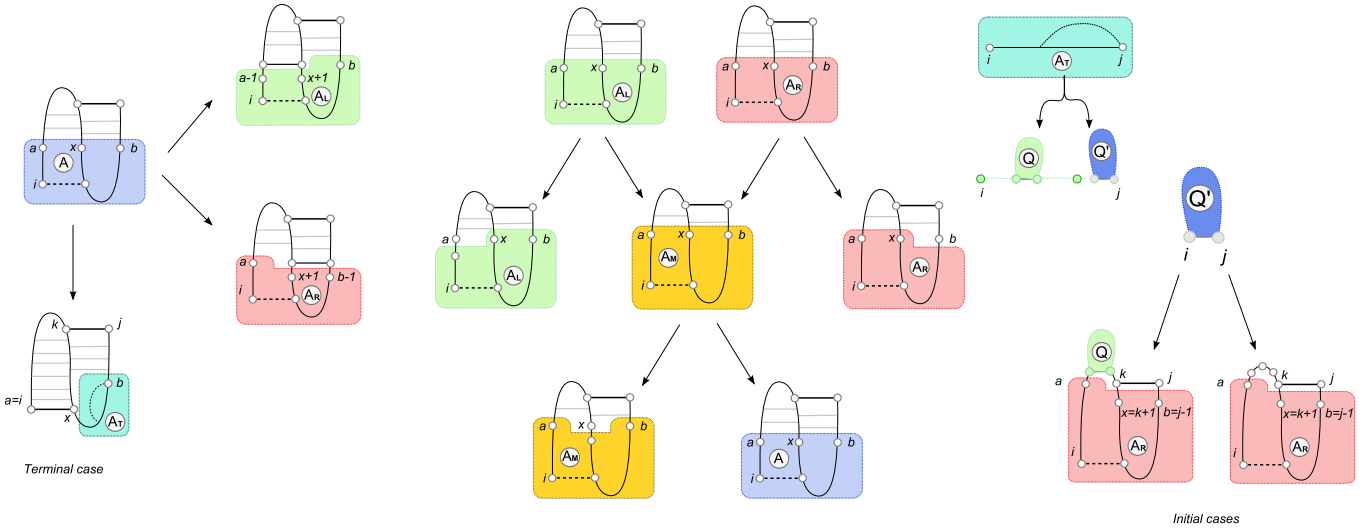
This decomposition is provably unambiguous and complete, and yields an F-graph that is acyclic, independent, and has $\Theta(n^2)$ vertices and $\mathcal{O}(n^3)$ arcs.

Application	Algorithm	Time	Memory	Reference
Energy minimization	Min. score + $\pi_{\mathcal{T}}$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	[ZS81]
Partition function	π count + $e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	[McC90]
Base-pairing probabilities	Arc prob. + $e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	[McC90]
Statistical sampling (k-samples)	Rand. gen. + $e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$\mathcal{O}(n^3 + k \cdot n \log n)$	$\mathcal{O}(n^2)$	[DL03, Pon08]
Moments of energy (Mean, Var.)	Moments + $e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	[MMN05]
k-th moment of additive features	Moments + $e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$\mathcal{O}(k^3.n^3)$	$\mathcal{O}(k.n^2)$	–
Correlations of additive features	Moments + $e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^2)$	–
Generalized moments	Moments + $e^{\frac{-\pi_{\mathcal{T}}}{RT}}$	$\mathcal{O}(4^k.n^3)$	$\mathcal{O}(2^k.n^2)$	–

Basic pseudoknots (Akutsu)



Basic pseudoknots (Akutsu)



Unambiguous decomposition capturing simple pseudoknots Akutsu *et al.*
$\Rightarrow$ yields $\mathcal{O}(n^4)/\mathcal{O}(n^4)$ time/memory algorithms for nearest neighbor energy model, and $\mathcal{O}(n^5)$ time/$\mathcal{O}(n^4)$ memory for Turner model.

We can now answer new questions:

- What probability for the MFE within Akutsu's class of conformations?
- What is the expected number of pseudoknots within a given sequence?
- ...

Conclusion

Using Hypergraphs, we were able to successfully detach conformation space from application.

- Implementation? We still have to mess with indices :(
  $\Rightarrow$ Start from CFG? Limited... (and already done by ADP) but maybe worthy
  $\Rightarrow$ Use Mathias Möhl's *split types*?
- Account for additional parameters (RNAMutants, RNABor...)
- Pseudoknots gene scanning (in progress). Non-canonical motifs?
- Rebuild distributions from truncated moments ($\Rightarrow$ Economics??)

References I

A. Condon, B. Davy, B. Rastegari, S. Zhao, and F. Tarrant.

Classifying RNA pseudoknotted structures.
Theoretical Computer Science, 320(1):35–50, 2004.

Y. Ding and E. Lawrence.

A statistical sampling algorithm for RNA secondary structure prediction.
Nucleic Acids Res, 31(24):7280–7301, 2003.

R. B. Lyngsø and C. N. S. Pedersen.

RNA pseudoknot prediction in energy-based models.
Journal of Computational Biology, 7(3-4):409–427, 2000.

N. Leontis and E. Westhof.

Geometric nomenclature and classification of RNA base pairs.
RNA, 7:499–512, 2001.

J.S. McCaskill.

The equilibrium partition function and base pair binding probabilities for RNA secondary structure.
Biopolymers, 29:1105–1119, 1990.

István Miklós, Irmtraud M Meyer, and Borbála Nagy.

Moments of the boltzmann distribution for rna secondary structures.
Bull Math Biol, 67(5):1031–1047, Sep 2005.

Y. Ponty.

Efficient sampling of RNA secondary structures from the boltzmann ensemble of low-energy: The boustrophedon method.
J Math Biol, 56(1-2):107–127, Jan 2008.

M. Zuker and P. Stiegler.

Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information.
Nucleic Acids Res, 9:133–148, 1981.

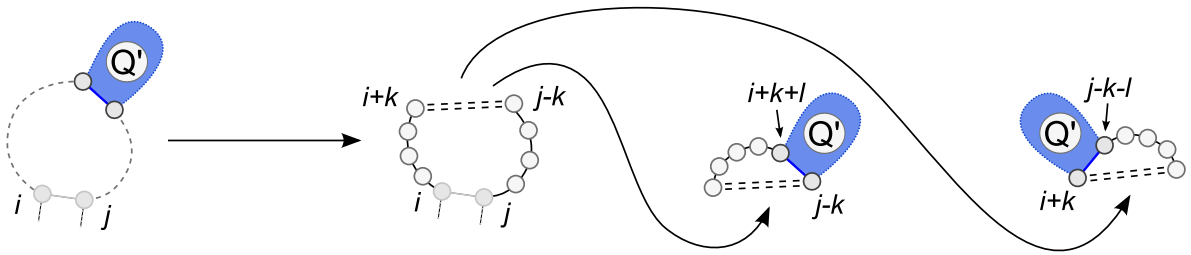


Figure: Alternative strategy for interior loops, creating the symmetric part first and the asymmetric part afterward.

Due to interior loops, the set F-arcs generated for the Q' case have apparent cardinality in $\mathcal{O}(n^4)$, while we claims $\mathcal{O}(n^3)$ complexities. A pragmatic answer may point out that it is common practice to upper bound the interior loop *size* $(j' - j) + (i' - i)$ to a predefined constant $K = 30$, bringing back the overall complexity to $\mathcal{O}(n^3)$. A more theoretically-sound approach achieves a similar complexity while remaining exact by further decomposing internal loops into a symmetric loop followed by a fully asymmetric one, as illustrated in Figure 1. Such a modification introduces a new case in the decomposition and captures with l the asymmetry of the loop, but the log nature of the entropy term in the Turner model then requires a small approximation.