Combinatorial descriptions of products in the category of forests and open order-preserving maps

Pietro Codara

Dipartimento di Informatica, Università degli Studi di Milano

(joint work with Ottavio M. D'Antona, and Vincenzo Marra)

Combinatorics 2014, Gaeta (LT) — June 3, 2014

Basic notions.

A category of forests

- A forest is a finite poset F such that for every $x \in F$, $\downarrow x$ is a chain. A tree is a forest with a bottom element.

A category of forests

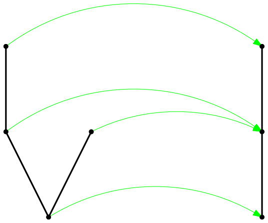
- A forest is a finite poset F such that for every $x \in F$, $\downarrow x$ is a chain. A tree is a forest with a bottom element.
- An order preserving map $f : F \longrightarrow G$ is open iff, for every $x \in F$,

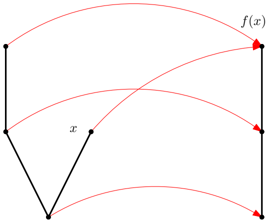
$$f(\downarrow x) = \downarrow f(x)\,.$$

A category of forests

- A forest is a finite poset F such that for every $x \in F$, $\downarrow x$ is a chain. A tree is a forest with a bottom element.
- An order preserving map $f : F \longrightarrow G$ is open iff, for every $x \in F$,

$$f(\downarrow x) = \downarrow f(x)\,.$$

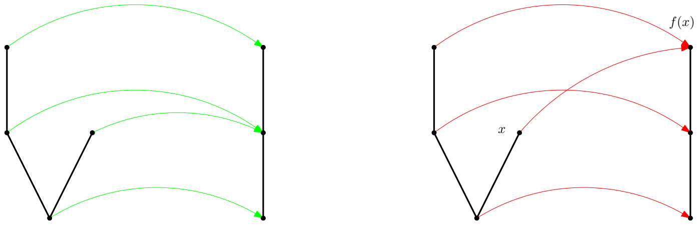




A category of forests

- A forest is a finite poset F such that for every $x \in F$, $\downarrow x$ is a chain. A tree is a forest with a bottom element.
- An order preserving map $f : F \longrightarrow G$ is open iff, for every $x \in F$,

$$f(\downarrow x) = \downarrow f(x).$$



(The category of forests and open maps is dually equivalent to the category of finitely presented Gödel algebras.)

Ordered partitions, and merged shuffles

- An ordered partition σ is a sequence of pairwise disjoint nonempty sets.

Ordered partitions, and merged shuffles

- An ordered partition σ is a sequence of pairwise disjoint nonempty sets.
- Given ordered partitions $\sigma = \{A_1, \ldots, A_m\}$, and $\tau = \{B_1, \ldots, B_n\}$ with $m \leq n$ we write $\sigma \leq \tau$ iff $A_i = B_i$ for every $i \in \{1, \ldots, m\}$.

Ordered partitions, and merged shuffles

- An ordered partition σ is a sequence of pairwise disjoint nonempty sets.
- Given ordered partitions $\sigma = \{A_1, \ldots, A_m\}$, and $\tau = \{B_1, \ldots, B_n\}$ with $m \leq n$ we write $\sigma \leq \tau$ iff $A_i = B_i$ for every $i \in \{1, \ldots, m\}$.
- Let σ and τ be ordered partitions with disjoint supports. An ordered partition θ is a shuffle of σ and τ iff σ and τ are subsequences of θ, and $\mathrm{supp}\theta = \mathrm{supp}\sigma \cup \mathrm{supp}\tau$;

Ordered partitions, and merged shuffles

- An ordered partition σ is a sequence of pairwise disjoint nonempty sets.
- Given ordered partitions $\sigma = \{A_1, \ldots, A_m\}$, and $\tau = \{B_1, \ldots, B_n\}$ with $m \leq n$ we write $\sigma \leq \tau$ iff $A_i = B_i$ for every $i \in \{1, \ldots, m\}$.
- Let σ and τ be ordered partitions with disjoint supports. An ordered partition θ is a shuffle of σ and τ iff σ and τ are subsequences of θ, and $\mathrm{supp}\theta = \mathrm{supp}\sigma \cup \mathrm{supp}\tau$;
- a merged shuffle is obtained by a shuffle θ, by merging some consecutive pairs of blocks $A, B \in \theta$, with $A \in \sigma$, and $B \in \tau$.

Ordered partitions, and merged shuffles

- An ordered partition σ is a sequence of pairwise disjoint nonempty sets.
- Given ordered partitions $\sigma = \{A_1, \ldots, A_m\}$, and $\tau = \{B_1, \ldots, B_n\}$ with $m \leq n$ we write $\sigma \leq \tau$ iff $A_i = B_i$ for every $i \in \{1, \ldots, m\}$.
- Let σ and τ be ordered partitions with disjoint supports. An ordered partition θ is a shuffle of σ and τ iff σ and τ are subsequences of θ, and $\mathrm{supp}\theta = \mathrm{supp}\sigma \cup \mathrm{supp}\tau$;
- a merged shuffle is obtained by a shuffle θ, by merging some consecutive pairs of blocks $A, B \in \theta$, with $A \in \sigma$, and $B \in \tau$.

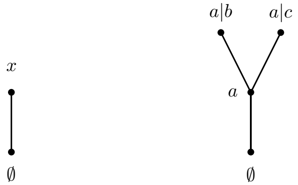
Example. Let $\sigma = \{a|b\}$ and $\tau = \{x\}$. The merged shuffles of σ and τ are: $\{a|b|x\}$, $\{a|x|b\}$, $\{x|a|b\}$, $\{a|bx\}$, $\{ax|b\}$.

Trees of ordered partitions

We can label trees with ordered partitions...

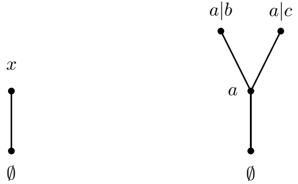
Trees of ordered partitions

We can label trees with ordered partitions...



Trees of ordered partitions

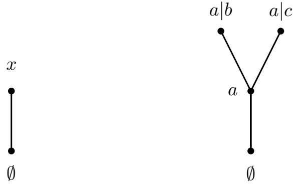
We can label trees with ordered partitions...



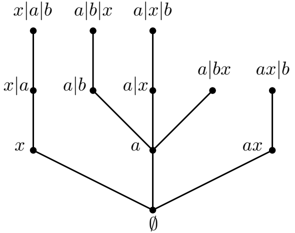
We can build a tree from a set of ordered partitions. The tree of merged shuffles of $\sigma = \{a|b\}$ and $\tau = \{x\}$ is...

Trees of ordered partitions

We can label trees with ordered partitions...



We can build a tree from a set of ordered partitions. The tree of merged shuffles of $\sigma = \{a|b\}$ and $\tau = \{x\}$ is...



Product of forests.

Product of forests

- Let $F = \{T_1, \ldots, T_r\}$ and $G = \{U_1, \ldots, U_s\}$ be forests. $F \times G = \{T_i \times U_j\}$, $i \in \{1, \ldots, r\}$, $j \in \{1, \ldots, s\}$.

Product of forests

- Let $F = \{T_1, \ldots, T_r\}$ and $G = \{U_1, \ldots, U_s\}$ be forests. $F \times G = \{T_i \times U_j\}$, $i \in \{1, \ldots, r\}$, $j \in \{1, \ldots, s\}$.
- The problem of describing $F \times G$ is reduced to that of describing its trees.

Product of forests

- Let $F = \{T_1, \ldots, T_r\}$ and $G = \{U_1, \ldots, U_s\}$ be forests. $F \times G = \{T_i \times U_j\}$, $i \in \{1, \ldots, r\}$, $j \in \{1, \ldots, s\}$.
- The problem of describing $F \times G$ is reduced to that of describing its trees.

How to compute the product of trees?

Product of trees

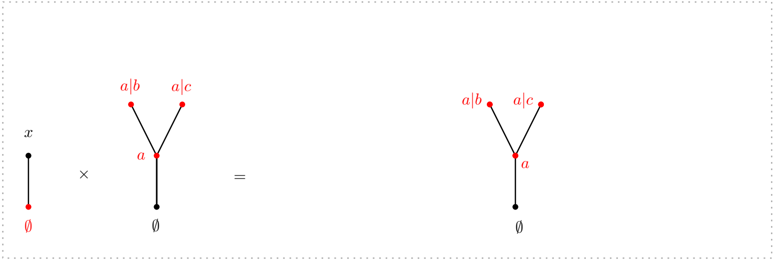
Computing the product of trees (an example).

[D'Antona, O.M., and Marra, V., *Computing coproducts of finitely presented Gödel algebras*, Ann. Pure Appl. Logic 142 (2006), 202–211]

Product of trees

Computing the product of trees (an example).

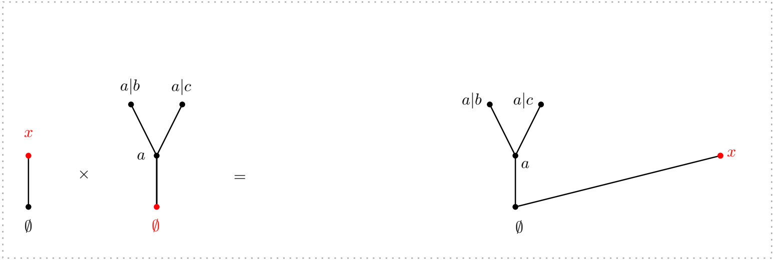
[D'Antona, O.M., and Marra, V., *Computing coproducts of finitely presented Gödel algebras*, Ann. Pure Appl. Logic 142 (2006), 202–211]



Product of trees

Computing the product of trees (an example).

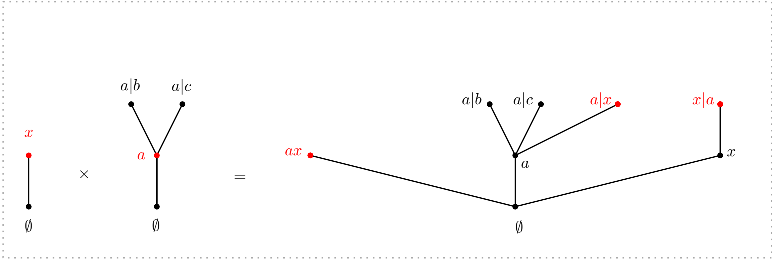
[D'Antona, O.M., and Marra, V., *Computing coproducts of finitely presented Gödel algebras*, Ann. Pure Appl. Logic 142 (2006), 202–211]



Product of trees

Computing the product of trees (an example).

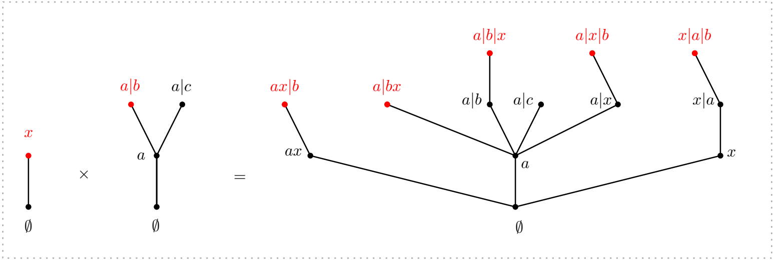
[D'Antona, O.M., and Marra, V., *Computing coproducts of finitely presented Gödel algebras*, Ann. Pure Appl. Logic 142 (2006), 202–211]



Product of trees

Computing the product of trees (an example).

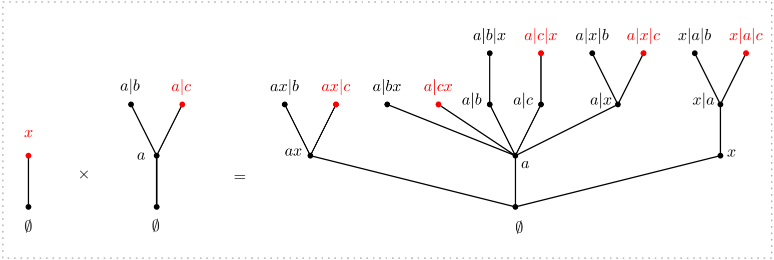
[D'Antona, O.M., and Marra, V., *Computing coproducts of finitely presented Gödel algebras*, Ann. Pure Appl. Logic 142 (2006), 202–211]



Product of trees

Computing the product of trees (an example).

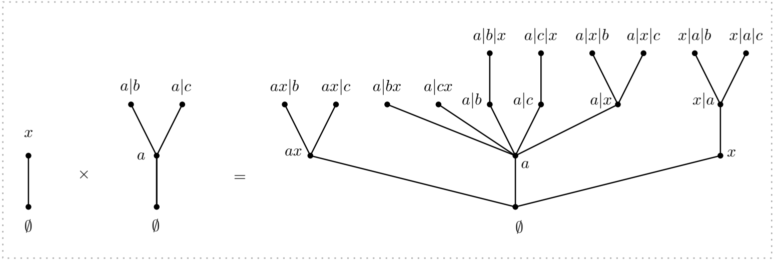
[D'Antona, O.M., and Marra, V., *Computing coproducts of finitely presented Gödel algebras*, Ann. Pure Appl. Logic 142 (2006), 202–211]



Product of trees

Computing the product of trees (an example).

[D'Antona, O.M., and Marra, V., *Computing coproducts of finitely presented Gödel algebras*, Ann. Pure Appl. Logic 142 (2006), 202–211]



Product of forests, an alternative way

Let F, G, and H be three forests.

Product of forests, an alternative way

Let F, G, and H be three forests.

- If $|F| = 1$, then $F \times G \cong G$.

Product of forests, an alternative way

Let F, G, and H be three forests.

- If $|F| = 1$, then $F \times G \cong G$.
- $(F + G) \times H \cong (F \times H) + (G \times H)$.

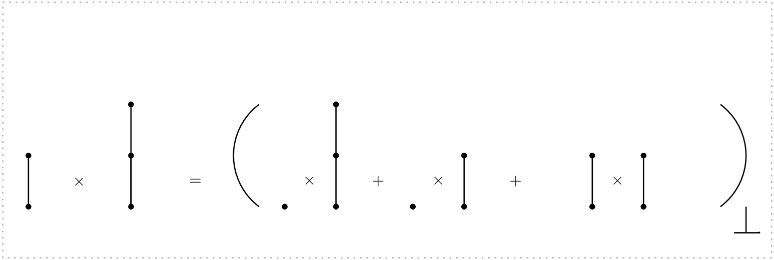
Product of forests, an alternative way

Let F, G, and H be three forests.

- If $|F| = 1$, then $F \times G \cong G$.
- $(F + G) \times H \cong (F \times H) + (G \times H)$.
- $F_\perp \times G_\perp \cong ((F \times G_\perp) + (F \times G) + (F_\perp \times G))_\perp$.

Product of trees, an alternative way

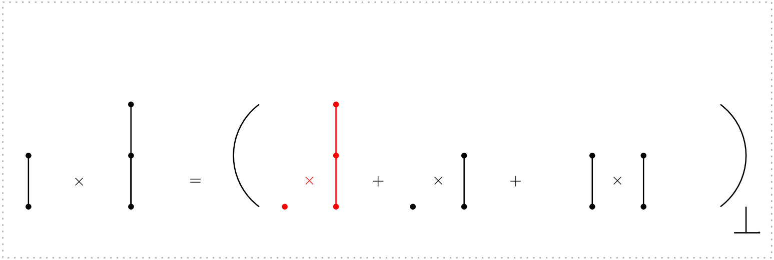
Computing the product of trees (an example).
[Aguzzoli, S., Bova, S., and Gerla, B., *Free Algebra and Functional Representation for Fuzzy Logics*, in Handbook of Mathematical Fuzzy Logic - Vol. 2, P. Cintula, P. Hájek, C. Noguera, eds., Studies in Logic, Vol. 38, College Pubblications, London (2011), 713–791]



Product of trees, an alternative way

Computing the product of trees (an example).

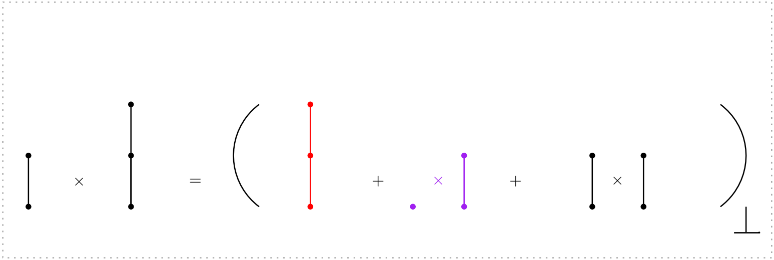
[Aguzzoli, S., Bova, S., and Gerla, B., *Free Algebra and Functional Representation for Fuzzy Logics*, in Handbook of Mathematical Fuzzy Logic - Vol. 2, P. Cintula, P. Hájek, C. Noguera, eds., Studies in Logic, Vol. 38, College Pubblications, London (2011), 713–791]



Product of trees, an alternative way

Computing the product of trees (an example).

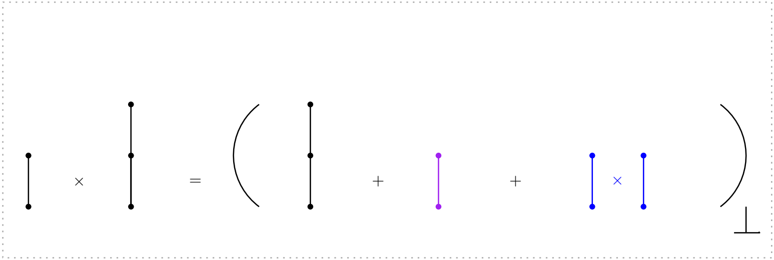
[Aguzzoli, S., Bova, S., and Gerla, B., *Free Algebra and Functional Representation for Fuzzy Logics*, in Handbook of Mathematical Fuzzy Logic - Vol. 2, P. Cintula, P. Hájek, C. Noguera, eds., Studies in Logic, Vol. 38, College Pubblications, London (2011), 713–791]



Product of trees, an alternative way

Computing the product of trees (an example).

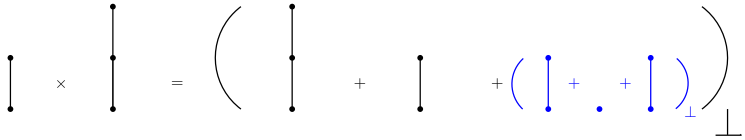
[Aguzzoli, S., Bova, S., and Gerla, B., *Free Algebra and Functional Representation for Fuzzy Logics*, in Handbook of Mathematical Fuzzy Logic - Vol. 2, P. Cintula, P. Hájek, C. Noguera, eds., Studies in Logic, Vol. 38, College Pubblications, London (2011), 713–791]



Product of trees, an alternative way

Computing the product of trees (an example).

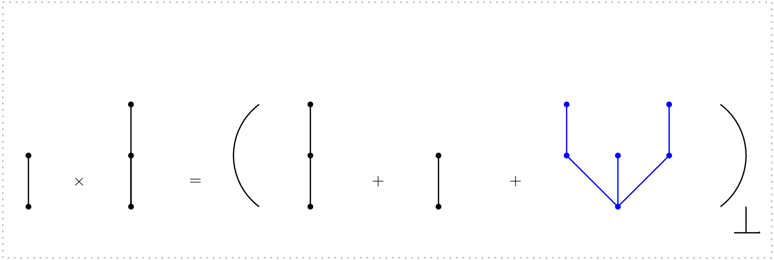
[Aguzzoli, S., Bova, S., and Gerla, B., *Free Algebra and Functional Representation for Fuzzy Logics*, in Handbook of Mathematical Fuzzy Logic - Vol. 2, P. Cintula, P. Hájek, C. Noguera, eds., Studies in Logic, Vol. 38, College Pubblications, London (2011), 713–791]



Product of trees, an alternative way

Computing the product of trees (an example).

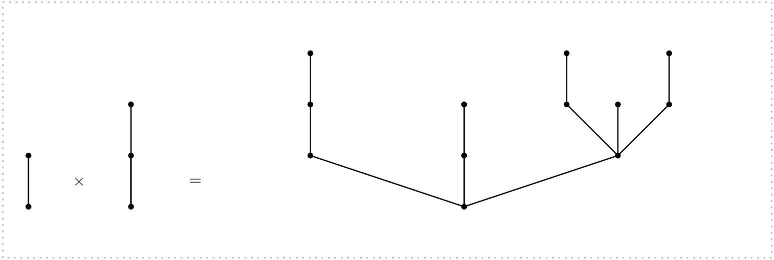
[Aguzzoli, S., Bova, S., and Gerla, B., *Free Algebra and Functional Representation for Fuzzy Logics*, in Handbook of Mathematical Fuzzy Logic - Vol. 2, P. Cintula, P. Hájek, C. Noguera, eds., Studies in Logic, Vol. 38, College Pubblications, London (2011), 713–791]



Product of trees, an alternative way

Computing the product of trees (an example).

[Aguzzoli, S., Bova, S., and Gerla, B., *Free Algebra and Functional Representation for Fuzzy Logics*, in Handbook of Mathematical Fuzzy Logic - Vol. 2, P. Cintula, P. Hájek, C. Noguera, eds., Studies in Logic, Vol. 38, College Pubblications, London (2011), 713–791]



Enumeration.

Delannoy numbers

The Delannoy number $D_{n,m}$ counts the number of lattice paths from $(0,0)$ to (n,m) in which only East, North, and Northeast steps are allowed. Delannoy numbers satisfy the following recurrence relation.

$$D_{n,m} = D_{n-1,m} + D_{n,m-1} + D_{n-1,m-1}$$

Delannoy numbers

The Delannoy number $D_{n,m}$ counts the number of lattice paths from $(0,0)$ to (n,m) in which only East, North, and Northeast steps are allowed. Delannoy numbers satisfy the following recurrence relation.

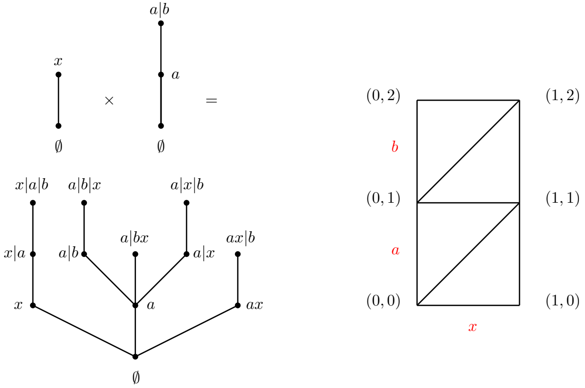
$$D_{n,m} = D_{n-1,m} + D_{n,m-1} + D_{n-1,m-1}$$

The following table shows some values of Delannoy numbers.

1	1	1	1	1	1	1	1
1	3	5	7	9	11	13	15
1	5	13	25	41	61	85	113
1	7	25	63	129	231	377	575
1	9	41	129	321	681	1289	2241
1	11	61	231	681	1683	3653	7183

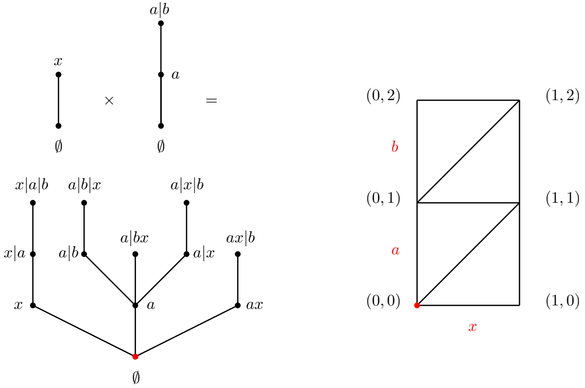
Delannoy numbers and ordered partitions, a simple case

In a simple case, one can associate each element of the product with a Delannoy path, as follows.



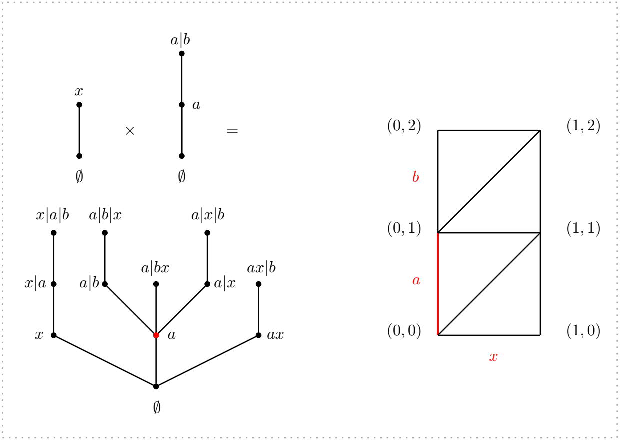
Delannoy numbers and ordered partitions, a simple case

In a simple case, one can associate each element of the product with a Delannoy path, as follows.



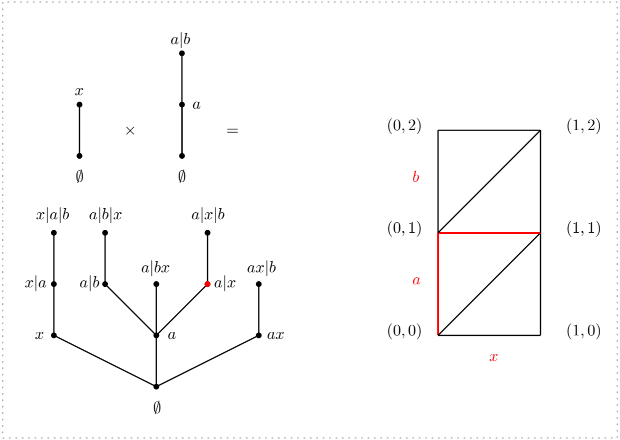
Delannoy numbers and ordered partitions, a simple case

In a simple case, one can associate each element of the product with a Delannoy path, as follows.



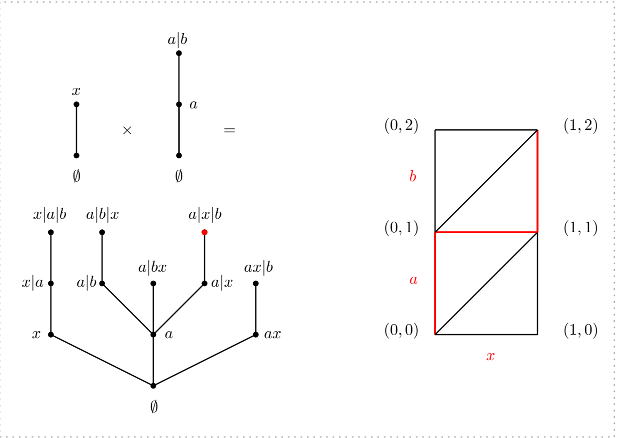
Delannoy numbers and ordered partitions, a simple case

In a simple case, one can associate each element of the product with a Delannoy path, as follows.



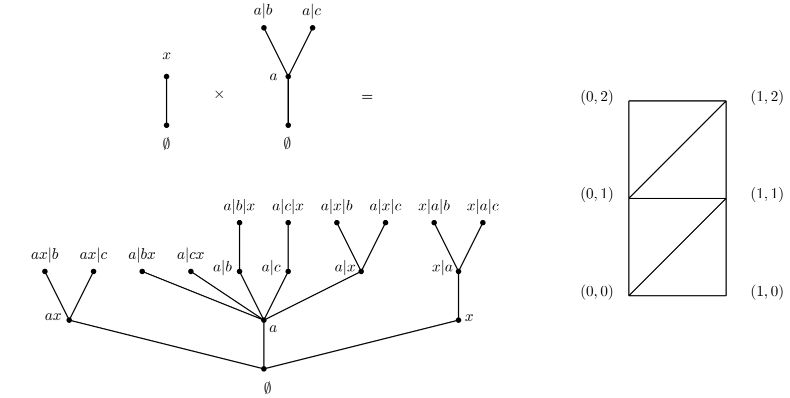
Delannoy numbers and ordered partitions, a simple case

In a simple case, one can associate each element of the product with a Delannoy path, as follows.



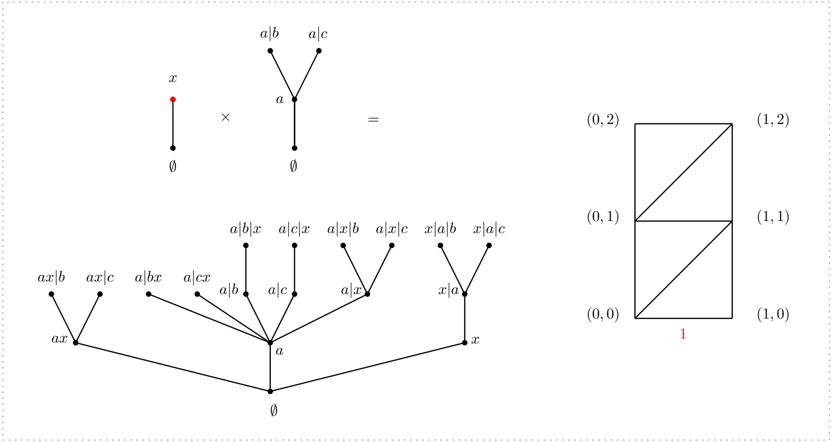
Delannoy numbers and ordered partitions, the general case

In general, one can add *multiplicity labels* to Delannoy steps in order to associate each Delannoy path to multiple elements of the product, as follows.



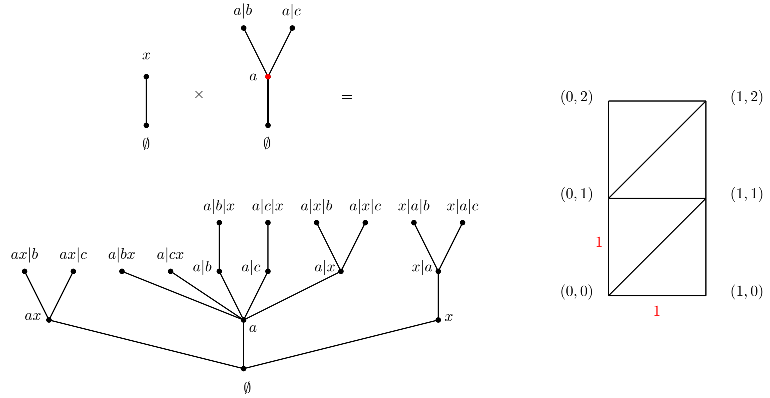
Delannoy numbers and ordered partitions, the general case

In general, one can add *multiplicity labels* to Delannoy steps in order to associate each Delannoy path to multiple elements of the product, as follows.



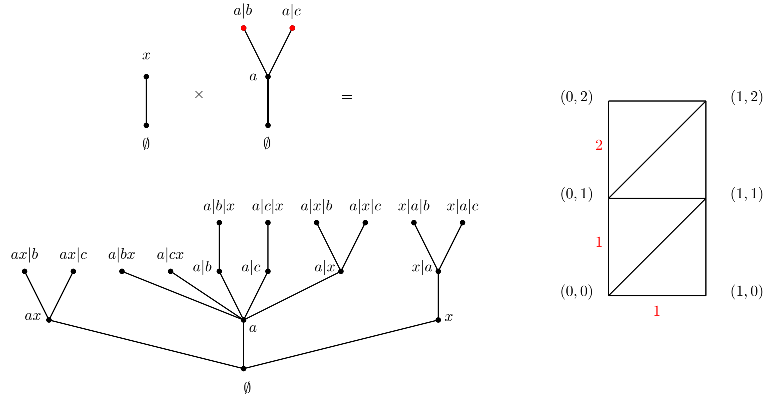
Delannoy numbers and ordered partitions, the general case

In general, one can add *multiplicity labels* to Delannoy steps in order to associate each Delannoy path to multiple elements of the product, as follows.



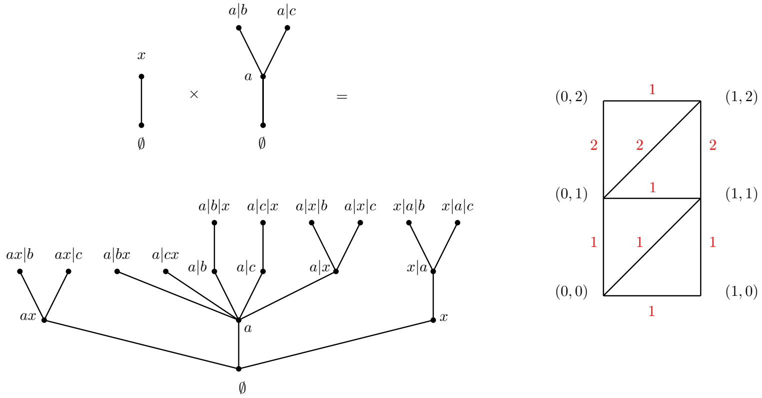
Delannoy numbers and ordered partitions, the general case

In general, one can add *multiplicity labels* to Delannoy steps in order to associate each Delannoy path to multiple elements of the product, as follows.



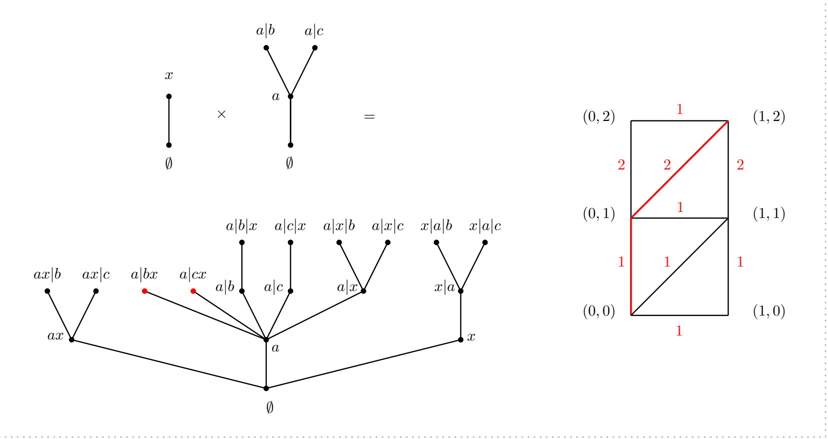
Delannoy numbers and ordered partitions, the general case

In general, one can add *multiplicity labels* to Delannoy steps in order to associate each Delannoy path to multiple elements of the product, as follows.



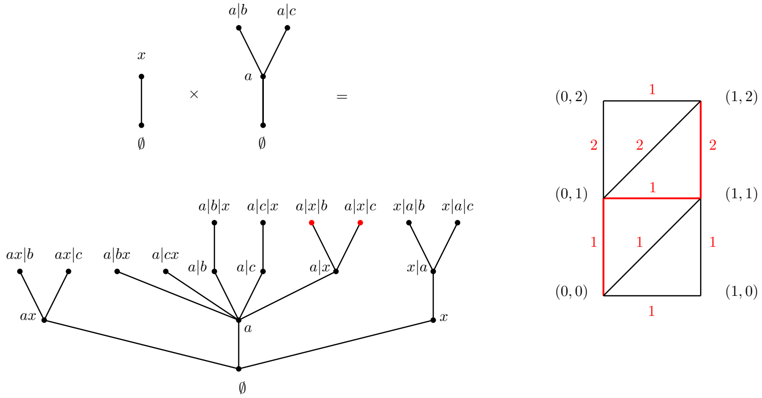
Delannoy numbers and ordered partitions, the general case

In general, one can add *multiplicity labels* to Delannoy steps in order to associate each Delannoy path to multiple elements of the product, as follows.



Delannoy numbers and ordered partitions, the general case

In general, one can add *multiplicity labels* to Delannoy steps in order to associate each Delannoy path to multiple elements of the product, as follows.



Formulæ, I

$$|T \times U| = \sum_{i \geq 0} \sum_{j \geq 0} t_i u_j D_{i,j} \,,$$

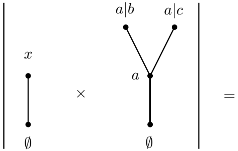
where t_i is the number of elements at level i of T, and u_j is the number of elements at level j of U.

Formulæ, I

$$|T \times U| = \sum_{i \geq 0} \sum_{j \geq 0} t_i u_j D_{i,j} ,$$

where t_i is the number of elements at level i of T, and u_j is the number of elements at level j of U.

Example.



$$= 1 \cdot 1 \cdot D_{0,0} + 1 \cdot 1 \cdot D_{0,1} + 1 \cdot 2 \cdot D_{0,2} + 1 \cdot 1 \cdot D_{1,0} + 1 \cdot 1 \cdot D_{1,1} + 1 \cdot 2 \cdot D_{1,2} =$$
$$= 1 + 1 + 2 + 1 + 3 + 10 = 18.$$

Formulæ, II

Let L_T be the row vector of length $n = \mathtt{high}(T)$ containing the level numbers of the tree T. Let L_U be the column vector of height $m = \mathtt{high}(U)$ containing the level numbers of the tree U. Let $\mathcal{D}_{n,m}$ be the $n \times m$ matrix such that $\mathcal{D}[i,j] = D_{i,j}$, for each i, j. Then,

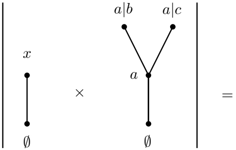
$$|T \times U| = L_T \times \mathcal{D}_{n,m} \times L_U \,.$$

Formulæ, II

Let L_T be the row vector of length $n = \mathtt{high}(T)$ containing the level numbers of the tree T. Let L_U be the column vector of height $m = \mathtt{high}(U)$ containing the level numbers of the tree U. Let $\mathcal{D}_{n,m}$ be the $n \times m$ matrix such that $\mathcal{D}[i,j] = D_{i,j}$, for each i, j. Then,

$$|T \times U| = L_T \times \mathcal{D}_{n,m} \times L_U \,.$$

Example.



$$\begin{bmatrix} 1 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 3 & 5 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = 18$$

Formulæ, III

Let $L_{T\times U}[h]$ be h^{th} level number of the product tree $T \times U$, for $h \geq 0$.

Formulæ, III

Let $L_{T\times U}[h]$ be h^{th} level number of the product tree $T \times U$, for $h \geq 0$. Clearly, $L_{T\times U}[0] = 1$.

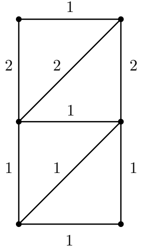
Formulæ, III

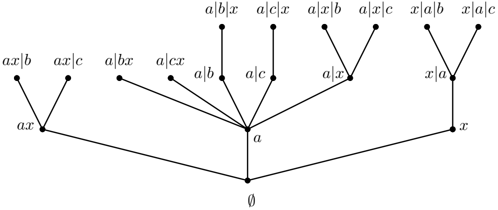
Let $L_{T\times U}[h]$ be h^{th} level number of the product tree $T \times U$, for $h \geq 0$. Clearly, $L_{T\times U}[0] = 1$. For $h > 1$, $L_{T\times U}[h]$ is the number of Delannoy paths, with their multiplicity, of length h contained in $(0,0) - (\mathtt{high}(T), \mathtt{high}(U))$.

Formulæ, III

Let $L_{T\times U}[h]$ be h^{th} level number of the product tree $T \times U$, for $h \geq 0$. Clearly, $L_{T\times U}[0] = 1$. For $h > 1$, $L_{T\times U}[h]$ is the number of Delannoy paths, with their multiplicity, of length h contained in $(0,0) - (\mathtt{high}(T), \mathtt{high}(U))$.

Example.

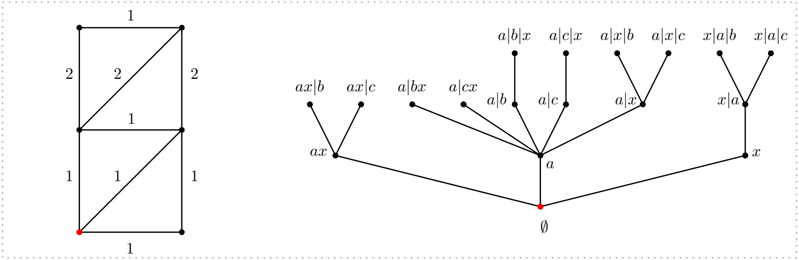




Formulæ, III

Let $L_{T\times U}[h]$ be h^{th} level number of the product tree $T \times U$, for $h \geq 0$. Clearly, $L_{T\times U}[0] = 1$. For $h > 1$, $L_{T\times U}[h]$ is the number of Delannoy paths, with their multiplicity, of length h contained in $(0,0) - (\mathtt{high}(T), \mathtt{high}(U))$.

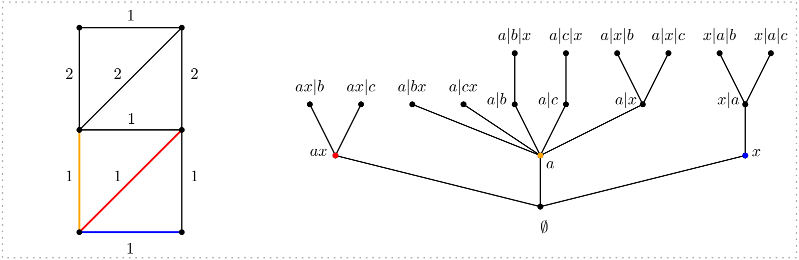
Example.



Formulæ, III

Let $L_{T\times U}[h]$ be h^{th} level number of the product tree $T \times U$, for $h \geq 0$. Clearly, $L_{T\times U}[0] = 1$. For $h > 1$, $L_{T\times U}[h]$ is the number of Delannoy paths, with their multiplicity, of length h contained in $(0,0) - (\mathtt{high}(T), \mathtt{high}(U))$.

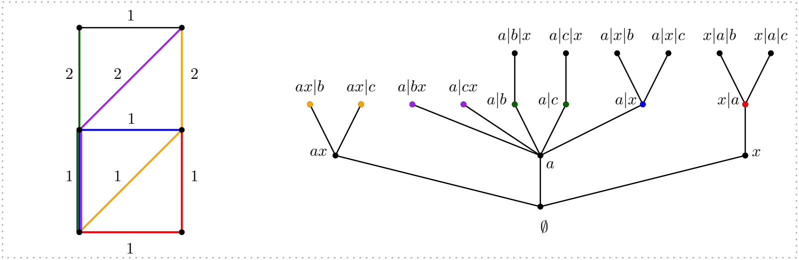
Example.



Formulæ, III

Let $L_{T\times U}[h]$ be h^{th} level number of the product tree $T \times U$, for $h \geq 0$. Clearly, $L_{T\times U}[0] = 1$. For $h > 1$, $L_{T\times U}[h]$ is the number of Delannoy paths, with their multiplicity, of length h contained in $(0,0) - (\mathtt{high}(T), \mathtt{high}(U))$.

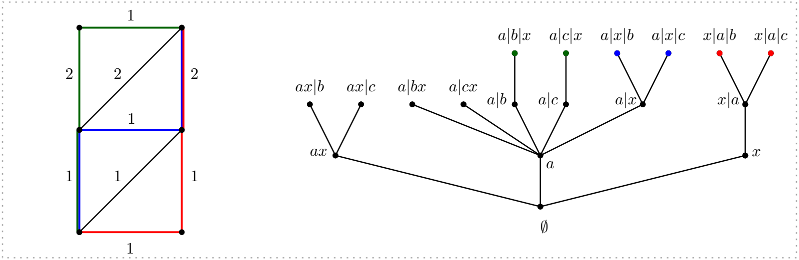
Example.



Formulæ, III

Let $L_{T\times U}[h]$ be h^{th} level number of the product tree $T \times U$, for $h \geq 0$. Clearly, $L_{T\times U}[0] = 1$. For $h > 1$, $L_{T\times U}[h]$ is the number of Delannoy paths, with their multiplicity, of length h contained in $(0,0) - (\mathtt{high}(T), \mathtt{high}(U))$.

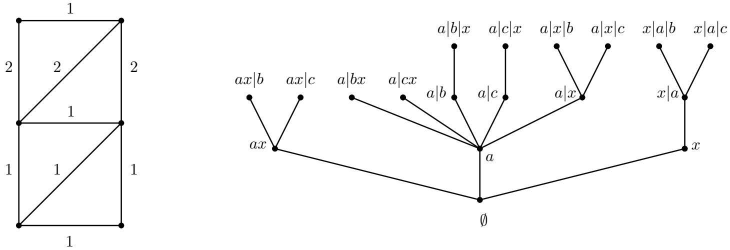
Example.



Formulæ, IV

Finally, we can count the number of leaves of the three $T \times U$ after a certain number p of *pruning*, as in the following example.

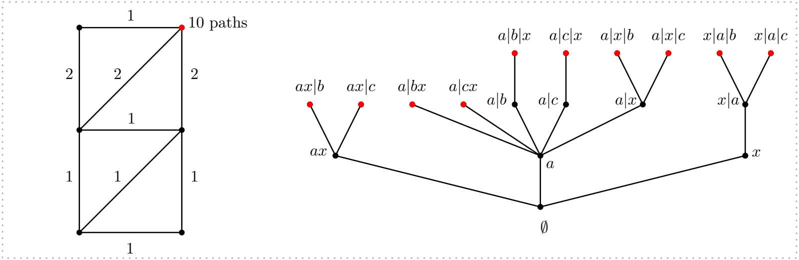
Example.



Formulæ, IV

Finally, we can count the number of leaves of the three $T \times U$ after a certain number p of *pruning*, as in the following example.

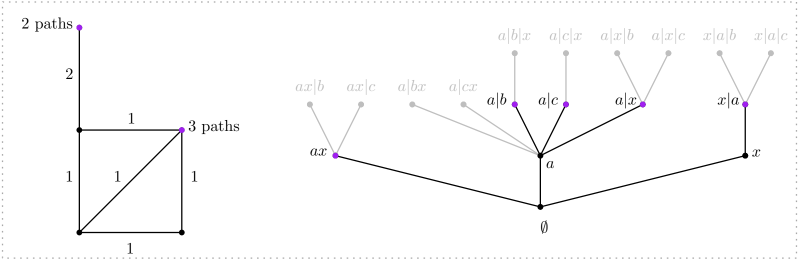
Example.



Formulæ, IV

Finally, we can count the number of leaves of the three $T \times U$ after a certain number p of *pruning*, as in the following example.

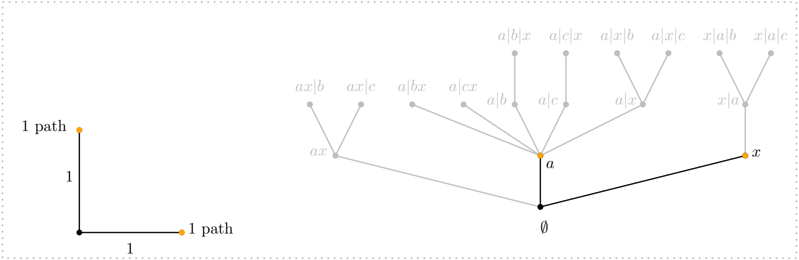
Example.



Formulæ, IV

Finally, we can count the number of leaves of the three $T \times U$ after a certain number p of *pruning*, as in the following example.

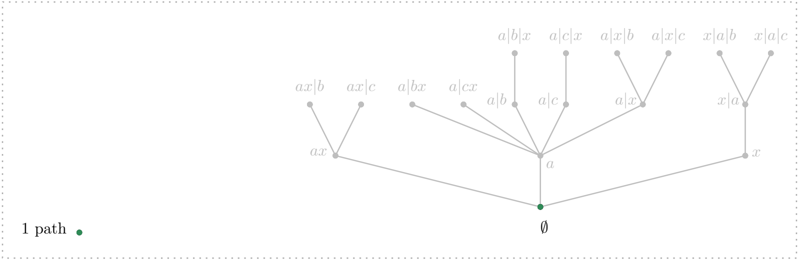
Example.



Formulæ, IV

Finally, we can count the number of leaves of the three $T \times U$ after a certain number p of *pruning*, as in the following example.

Example.



Thank you for your attention.