

Artificial Intelligence

Cours5 - Version Space Learning

L3 - Informatique

Nadjib Lazaar

Ing - Phd - HDR - Professor - Paris-Saclay University - LISN - LaHDAK

lazaar@lisn.fr

<https://perso.lisn.upsaclay.fr/lazaar/>

24/01/2025

Version Space Learning

Introduction

- **Définition du Version Space Learning**
 - Proposé par **Tom Mitchell (1978)**
 - Apprentissage supervisé basé sur l'élimination des hypothèses
 - Recherche d'un **ensemble de concepts compatibles** avec les exemples
- **Pourquoi l'utiliser ?**
 - Approche explicable et interprétable
 - Nécessite peu de données comparé aux réseaux de neurones
 - Permet une généralisation contrôlée

Version Space Learning

Contexte et Historique

- **Deux grandes approches en Machine Learning**
 - Connexionniste → Réseaux de neurones (apprentissage statistique)
 - Symbolique → Manipulation de concepts explicites (ex : VSL)
- **Objectif du Version Space Learning**
 - Identifier une règle logique qui sépare exemples positifs et exemples négatifs
 - Manipuler directement des concepts explicites

Version Space Learning

ML Connexionniste vs ML Symbolique

Caractéristique	ML Connexionniste	ML Symbolique
Représentation	Réseaux de neurones, poids synaptiques	Règles logiques, structures symboliques
Interprétabilité	Boîte noire, difficile à expliquer	Explicable, concepts clairs
Données nécessaires	Gros volumes, besoin d'annotations	Peut fonctionner avec peu d'exemples
Approche	Approximative, probabiliste	Déductive, basée sur des règles
Capacité d'abstraction	Apprend des patterns statistiques	Peut généraliser avec peu d'exemples
Besoin de connaissances	Non	Oui, souvent

Concept Learning

Définition du Problème

- **Qu'est-ce que l'apprentissage de concepts ?**
 - Une approche d'apprentissage automatique pour identifier une règle de décision
 - **Objectif** : distinguer les instances positives des instances négatives
- **Formulation mathématique :**
- $f : X \rightarrow \{0,1\}$:
 - X : espace des instances
 - $f(x) = 1$: instance positive
 - $f(x) = 0$: instance négative
 - L'ensemble des instances positives est défini comme : $C = \{x \in X \mid f(x) = 1\}$

Concept Learning

Définition du Problème

- **Qu'est-ce que l'apprentissage de concepts ?**
 - Une approche d'apprentissage automatique pour identifier une règle de décision
 - **Objectif** : distinguer les instances positives des instances négatives
- **Formulation mathématique :**
- $f : X \rightarrow \{0,1\}$:
 - X : espace des instances
 - $f(x) = 1$: instance positive
 - $f(x) = 0$: instance négative
- L'ensemble des instances positives est défini comme : $C = \{x \in X \mid f(x) = 1\}$

Nombre total de concepts possibles :
 $|C| = 2^{|X|}$

Concept Learning

Hypothèses et Contraintes

- **Principales hypothèses de l'apprentissage de concepts :**
 1. **Hypothèse du bruit :** Les exemples d'entraînement sont exacts
 2. **Hypothèse du concept bien défini :** Le concept cible est apprenable
 3. **Hypothèse du monde fermé :** Tout ce qui n'est pas mentionné comme positif est négatif
- Ces hypothèses peuvent être restrictives dans des contextes réels où les données peuvent contenir du bruit

Concept Learning

Illustration et Exemple

- **Exemple 1 : Reconnaissance des formes**
 - X : formes géométriques
 - C : ensemble des cercles
 - $X \setminus C$: autres formes (carrés, triangles, etc.)
- **Exemple 2 : Classification des emails**
 - X : ensemble des emails
 - C : emails considérés comme spam
 - $X \setminus C$: emails non-spam (ham)
 - Un algorithme apprend une fonction f qui prédit si un email est spam ou non en fonction de certaines caractéristiques.

Concept Learning

Lien avec les Hypothèses d'Apprentissage

- Facteurs influençant l'apprentissage de concepts :
 1. **Stationnarité** : Distribution stable des instances dans le temps
 2. **Représentativité** : Les exemples d'entraînement sont représentatifs
 3. **Absence d'ambiguïté** : Une instance appartient à une seule classe

Concept Learning

Ensemble des Hypothèses

- Une hypothèse h est une règle de décision permettant de classer les instances
- L'ensemble des hypothèses H représente toutes les règles possibles

Concept Learning

Exemple : Classification des formes géométriques

- **Attributs considérés:**

- **Couleur :** Bleu ou Rouge ($X_1 \in \{B, R\}$)
- **Forme :** Carré ou Cercle ($X_2 \in \{C, S\}$)
- **Taille :** Petit ou Grand ($X_3 \in \{P, G\}$)

- **Exemples d'hypothèses possibles :**

- h_1 : « Toutes les formes sont des carrés » $\Rightarrow h_1 \equiv (_, S, _)$
- h_2 : « Les formes grandes et rouges » $\Rightarrow h_2 \equiv (R, _, G)$
- h_3 : « Les cercles bleus et petits » $\Rightarrow h_3 \equiv (B, C, P)$
- L'ensemble des hypothèses H contient toutes les combinaisons possibles de ces règles

Version Space Learning

Définition de l'Espace de Version

- Qu'est-ce que l'espace de version ?

- Ensemble des hypothèses compatibles avec les exemples d'apprentissage
- Défini comme :

$$VS = \{h \in H \mid h(x) = f(x) \text{ pour tous les exemples } x \text{ observés}\}$$

- Où :
 - H : ensemble des hypothèses possibles
 - $h(x)$: prédiction d'une hypothèse h pour une instance x
 - $f(x)$: étiquette réelle de x

Version Space Learning

Ordre Général-Spécifique

- Organisation des hypothèses
- Une hypothèse h_1 est plus générale que h_2 si :

$$h_1 \geq h_2 \Leftrightarrow \forall x \in X, h_2(x) = 1 \Rightarrow h_1(x) = 1$$

Formes Carrés : $h_1 \equiv (_, S, _)$

Petites Formes Carrés $h_2 \equiv (_, S, P)$

- Cet ordre permet d'organiser les hypothèses en treillis

Structure de Treillis

Poset (S, R)

- Un ordre partiel est une relation binaire R sur un ensemble S :
 $\forall x, y, z \in S$
- $x R x$ (reflexivity)
- $x R y \wedge y R x \Rightarrow x = y$ (anti-symmetry)
- $x R y \wedge y R z \Rightarrow x R z$ (transitivity)

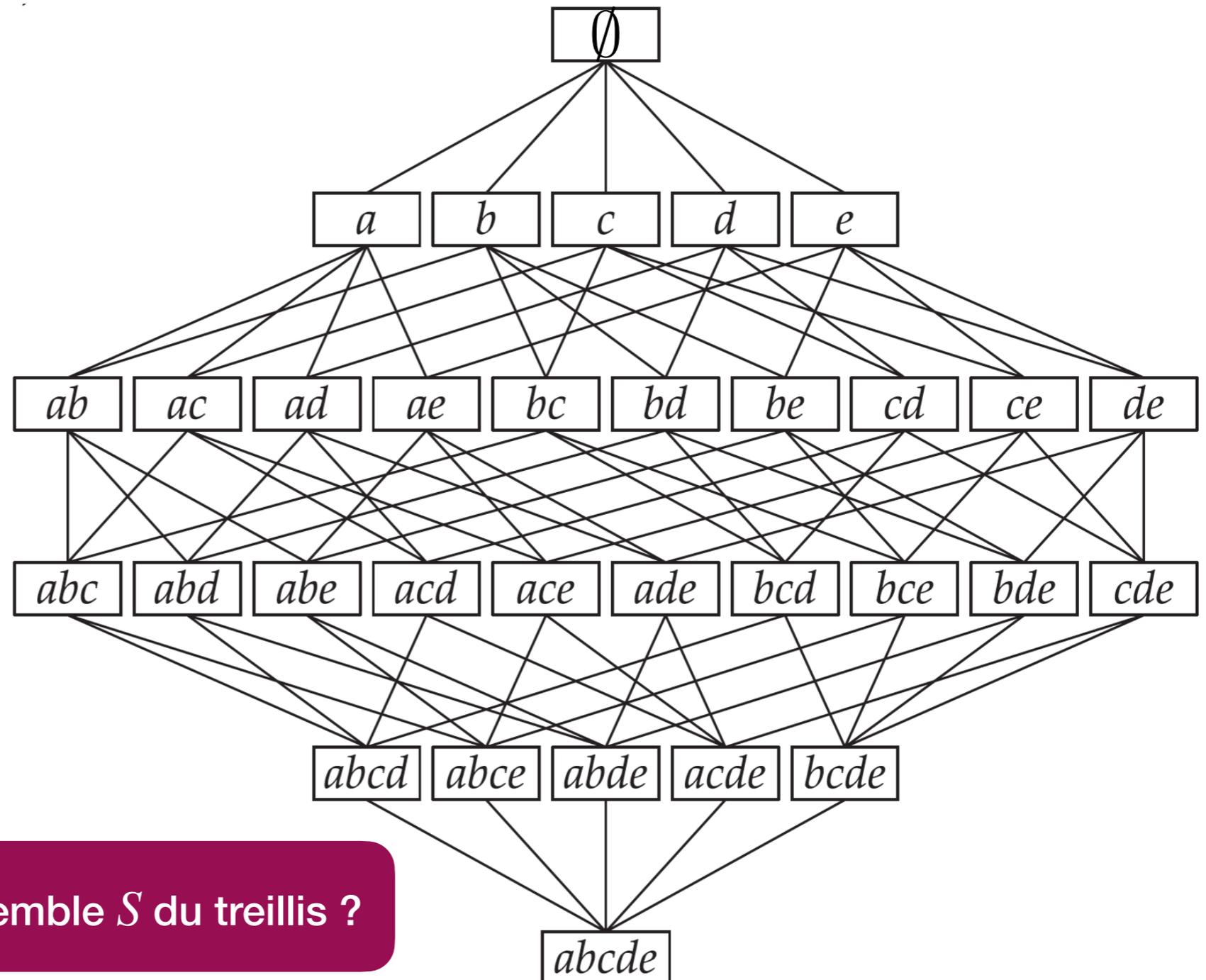
Structure de Treillis

Poset (S, R)

Quelle est la relation R et l'ensemble S du treillis ?

Structure de Treillis

Poset (S, R)



Quelle est la relation R et l'ensemble S du treillis ?

Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

X :

Couleur	Forme	Taille
---------	-------	--------

Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

X :

Couleur	Forme	Taille
---------	-------	--------



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

X :

Couleur	Forme	Taille
---------	-------	--------



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

X :

Couleur	Forme	Taille
---------	-------	--------

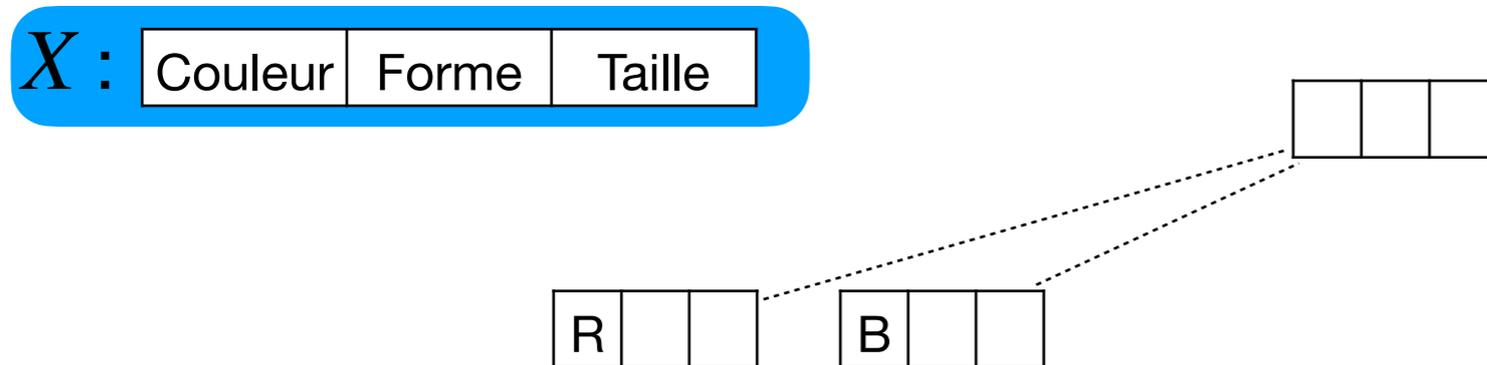
R		
---	--	--

--	--	--

∅	∅	∅
---	---	---

Version Space Learning

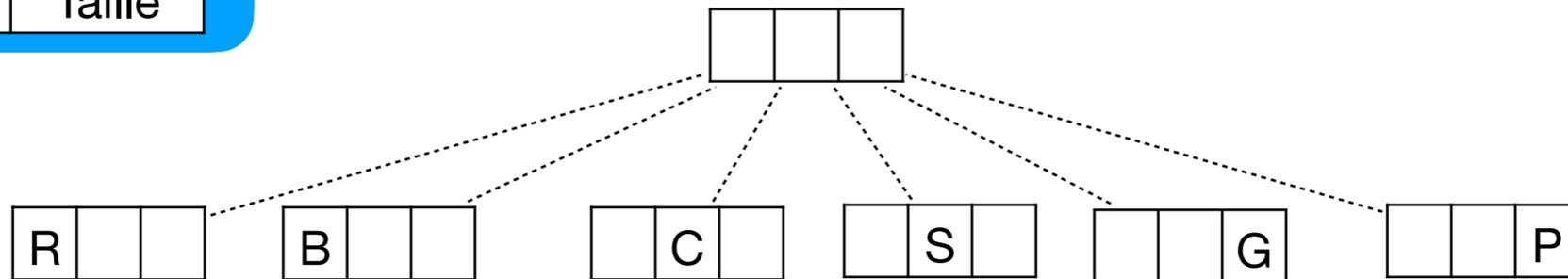
Exemple de Treillis (Reconnaissance des formes)



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

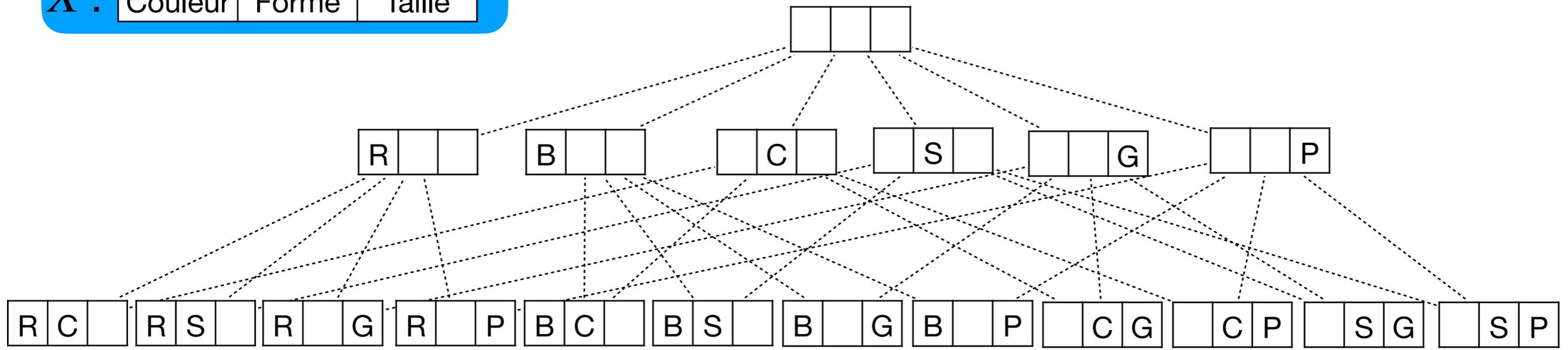
X : Couleur | Forme | Taille



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

X : Couleur | Forme | Taille

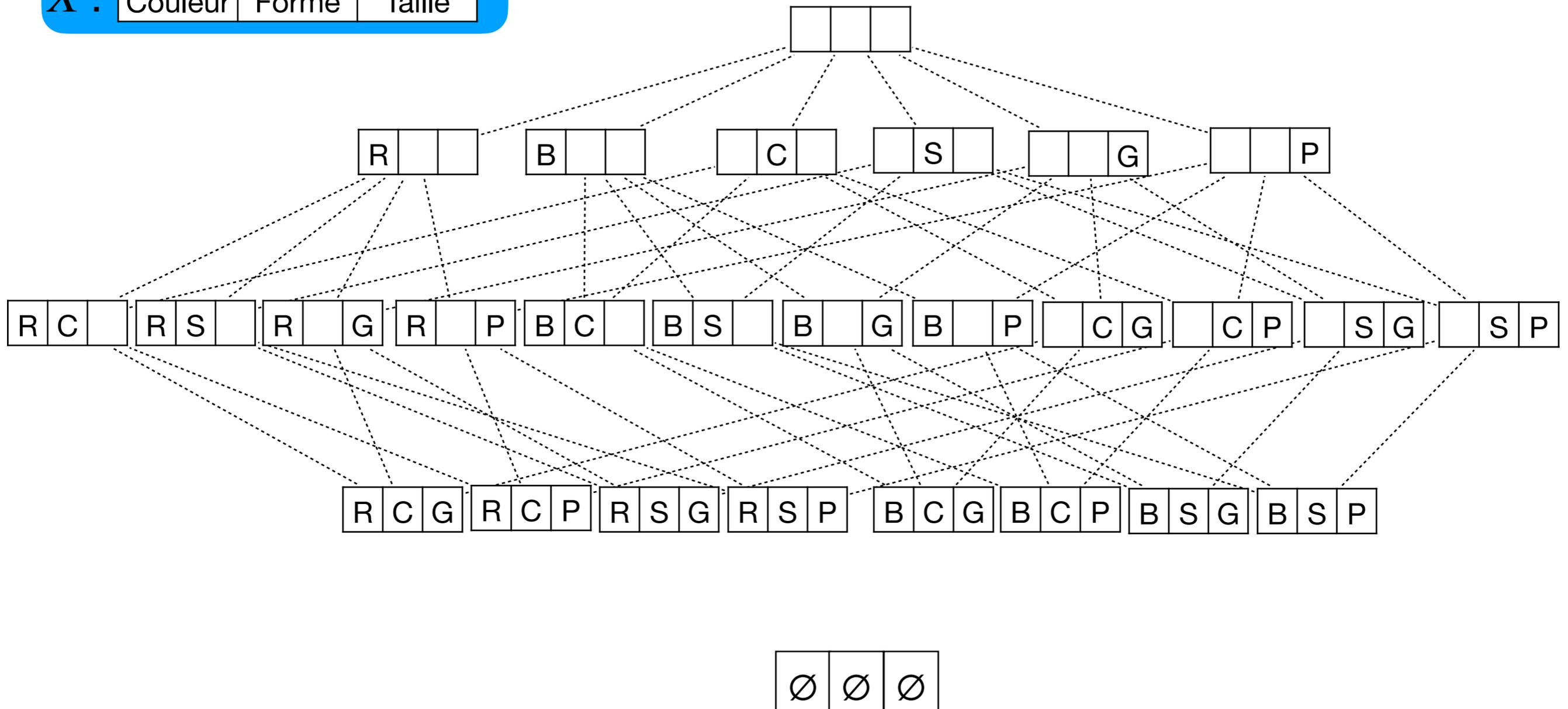


\emptyset \emptyset \emptyset

Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

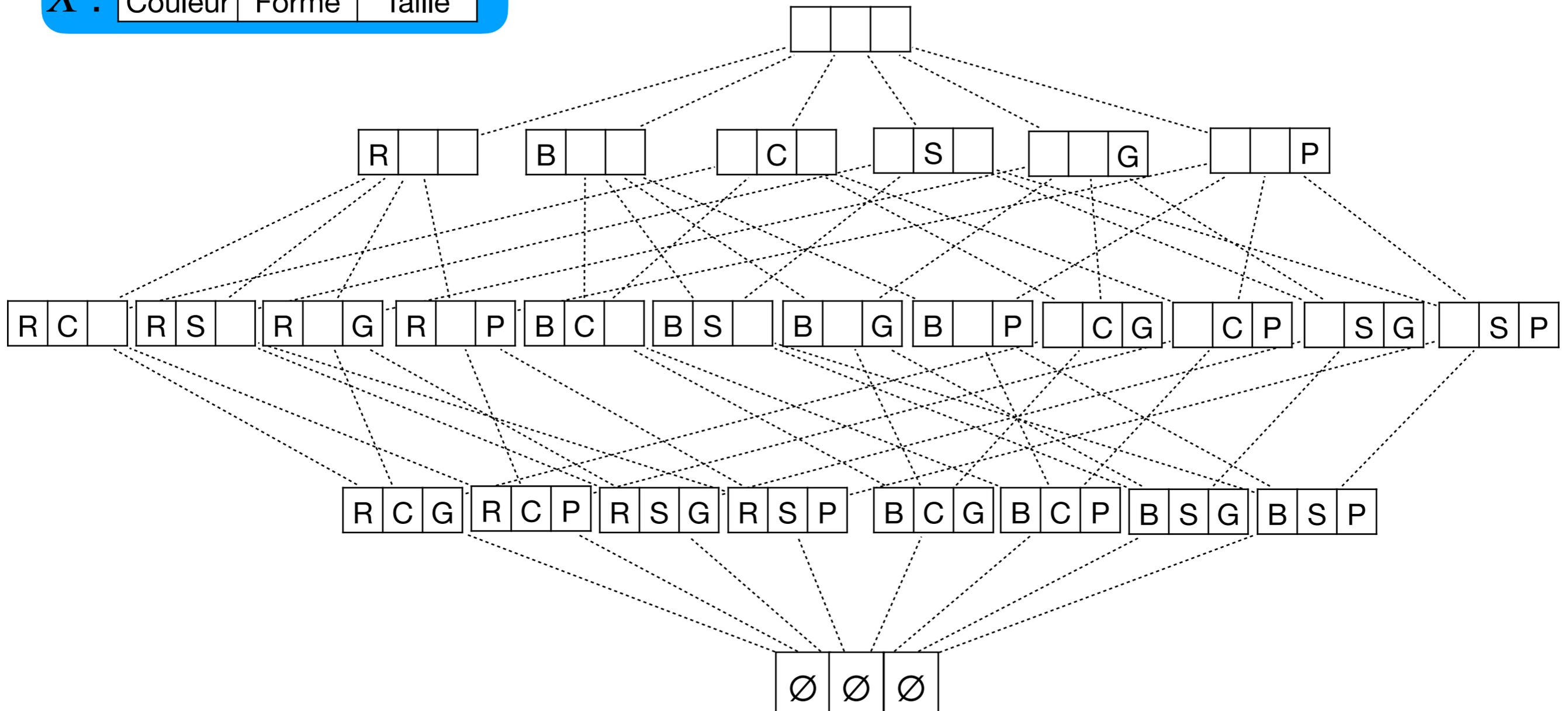
X : Couleur | Forme | Taille



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

X : Couleur | Forme | Taille



Version Space Learning

Bornes du Treillis

- Définition des ensembles limites :
- **G-set (général)** : hypothèses les plus générales compatibles :

$$G = \{h \in H \mid h(x) = 0, \quad \forall (x,0) \in D \text{ et } \nexists h' \in H, h \subset h'\}$$

- **S-set (spécifique)** : hypothèses les plus spécifiques compatibles :

$$S = \{h \in H \mid h(x) = 1, \quad \forall (x,1) \in D \text{ et } \nexists h' \in H, h' \subset h\}$$

- L'espace de version est compris entre ces bornes :

$$S \subseteq VS \subseteq G$$

Version Space Learning

Concept Learning et VSL

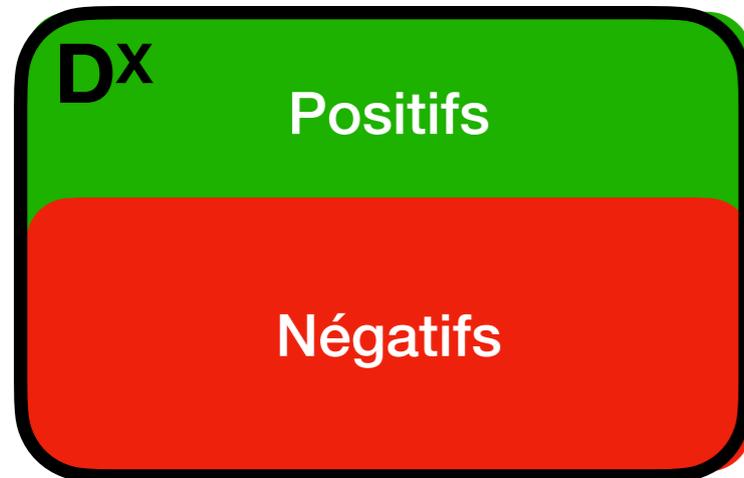
Version Space Learning

Concept Learning et VSL



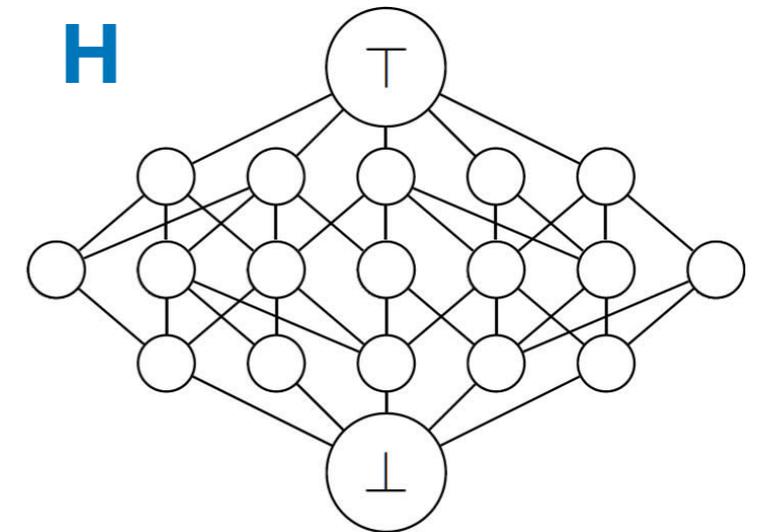
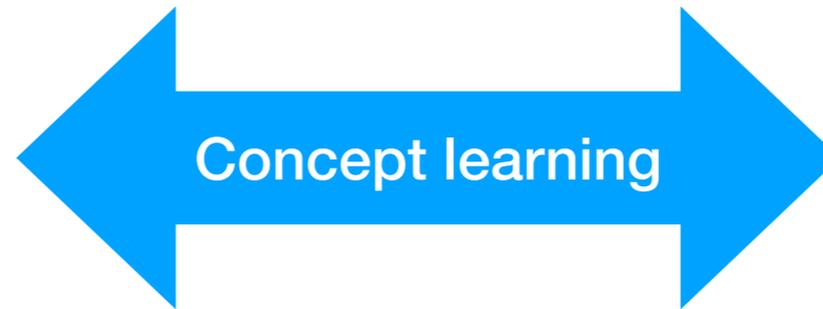
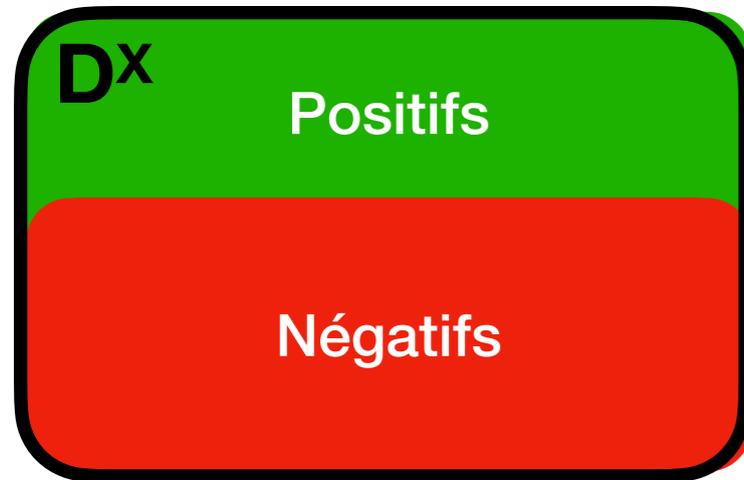
Version Space Learning

Concept Learning et VSL



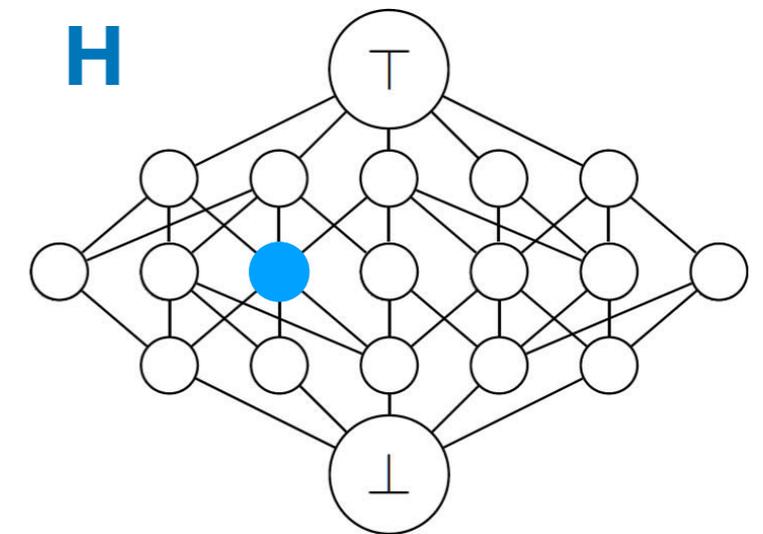
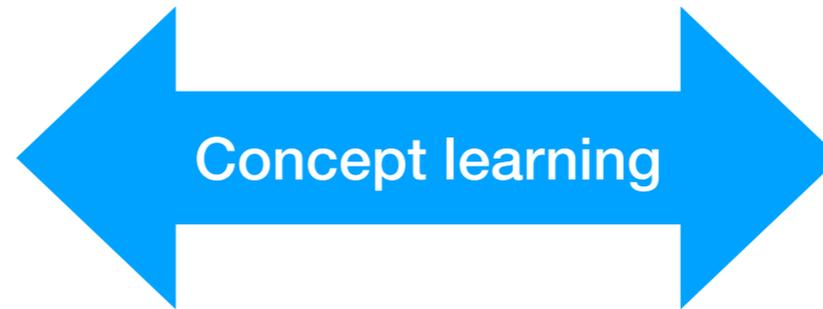
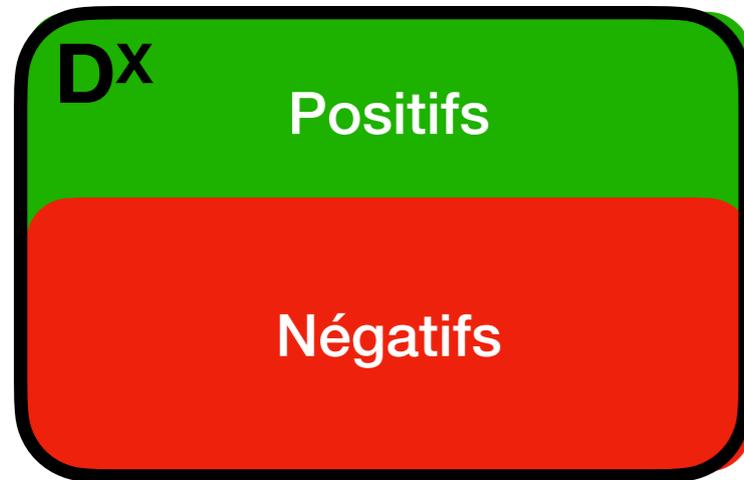
Version Space Learning

Concept Learning et VSL



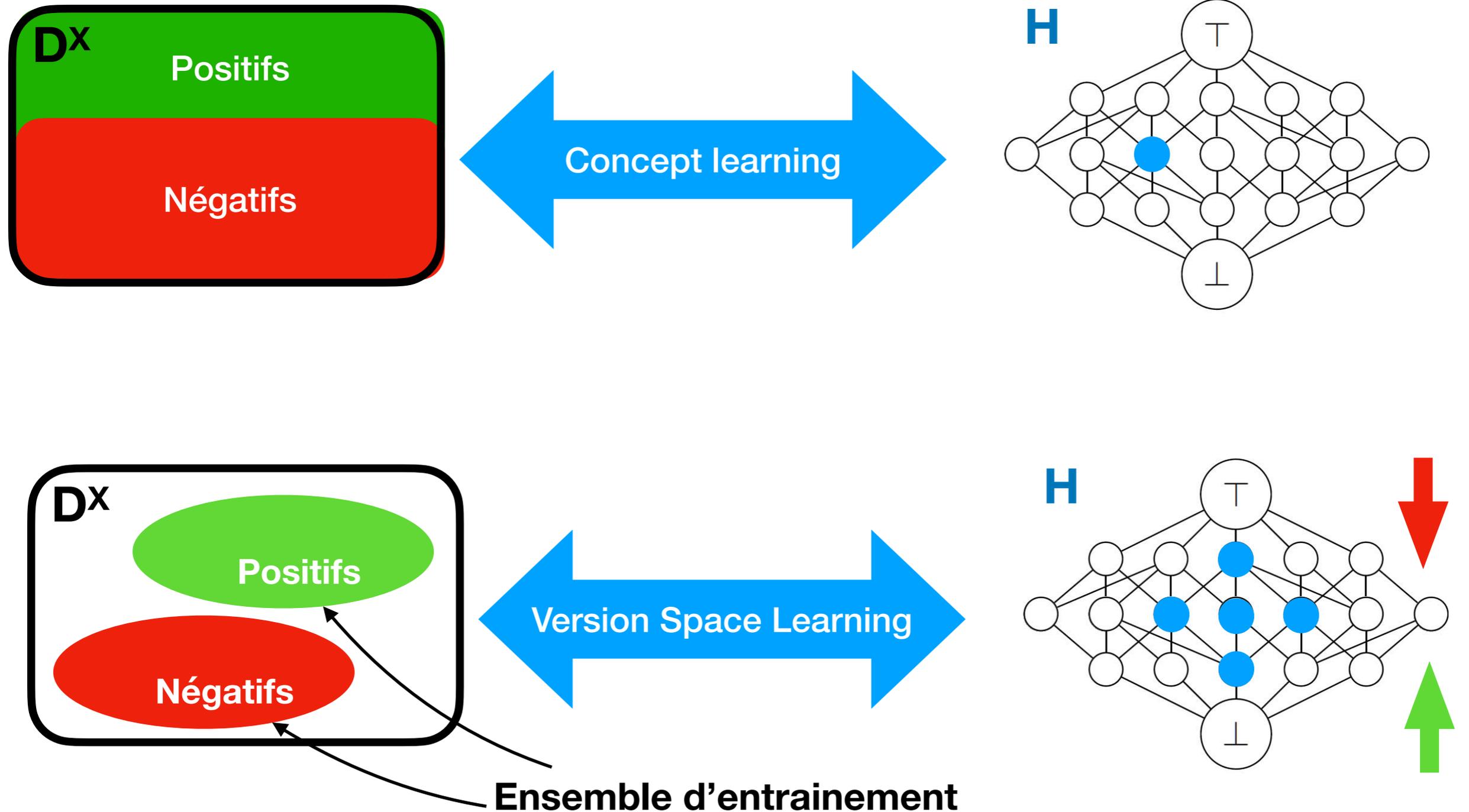
Version Space Learning

Concept Learning et VSL



Version Space Learning

Concept Learning et VSL



Version Space Learning

Algorithme

Algorithme 1 : Algorithme Version Space Learning

Input : Un ensemble d'exemples d'entraînement D avec attributs et classes

Output : Les ensembles S et G définissant l'espace des versions

Initialisation :

$S \leftarrow$ hypothèse la plus spécifique (ex : un ensemble vide ou une instance spécifique)

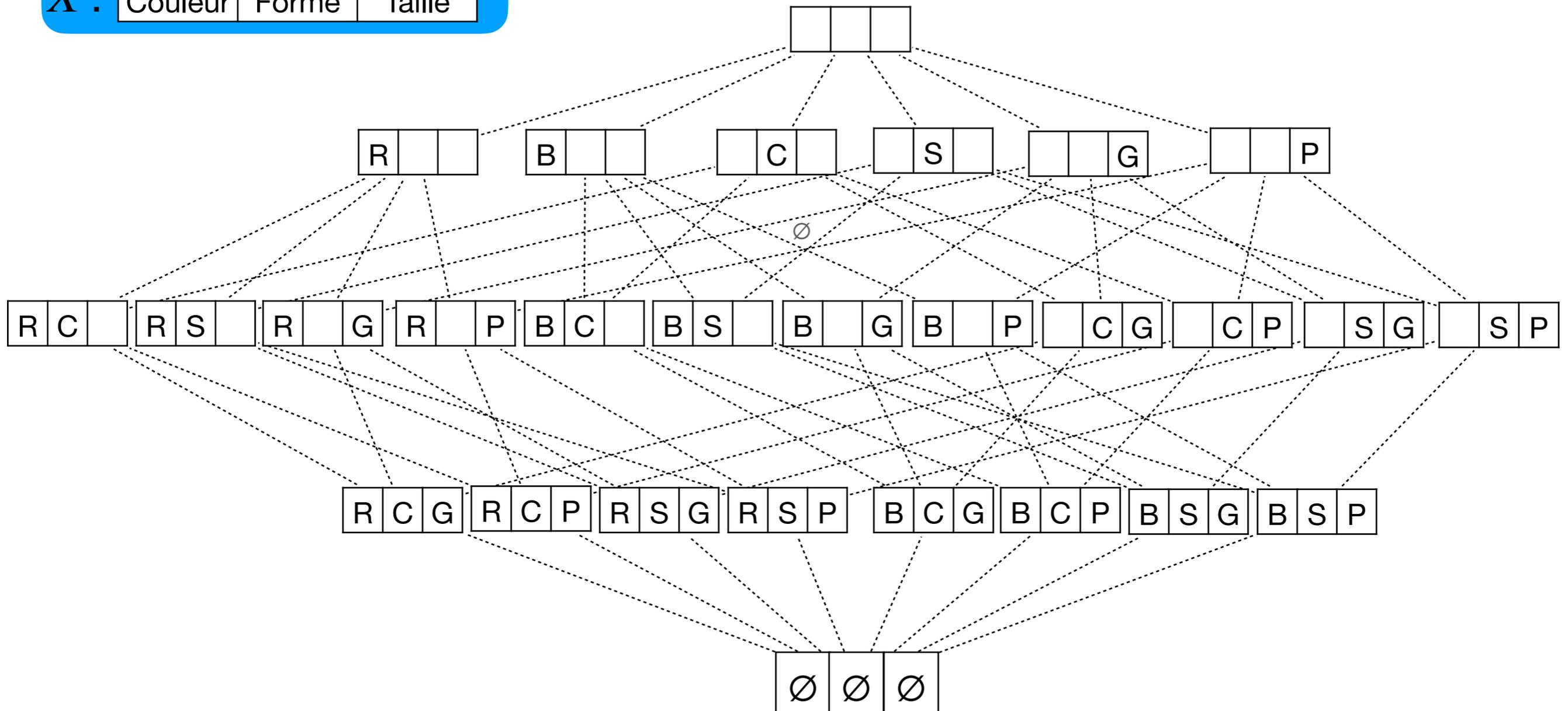
$G \leftarrow$ hypothèse la plus générale (ex : (?, ?, ..., ?))

```
foreach exemple  $(X, y) \in D$  do
  if y est positif (exemple accepté) then
    foreach  $g \in G$  do
      if g ne couvre pas X then
        | Retirer  $g$  de  $G$ 
      end
    end
    foreach  $s \in S$  do
      if s ne couvre pas X then
        | Généraliser  $s$  au minimum pour couvrir  $X$ 
      end
    end
    end
    Retirer de  $S$  les hypothèses plus générales que d'autres
  else if y est négatif (exemple rejeté) then
    foreach  $s \in S$  do
      if s couvre X then
        | Retirer  $s$  de  $S$ 
      end
    end
    foreach  $g \in G$  do
      if g couvre X then
        | Spécialiser  $g$  pour exclure  $X$ 
      end
    end
    end
    Retirer de  $G$  les hypothèses plus spécifiques que d'autres
  end
end
return  $S, G$ 
```

Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

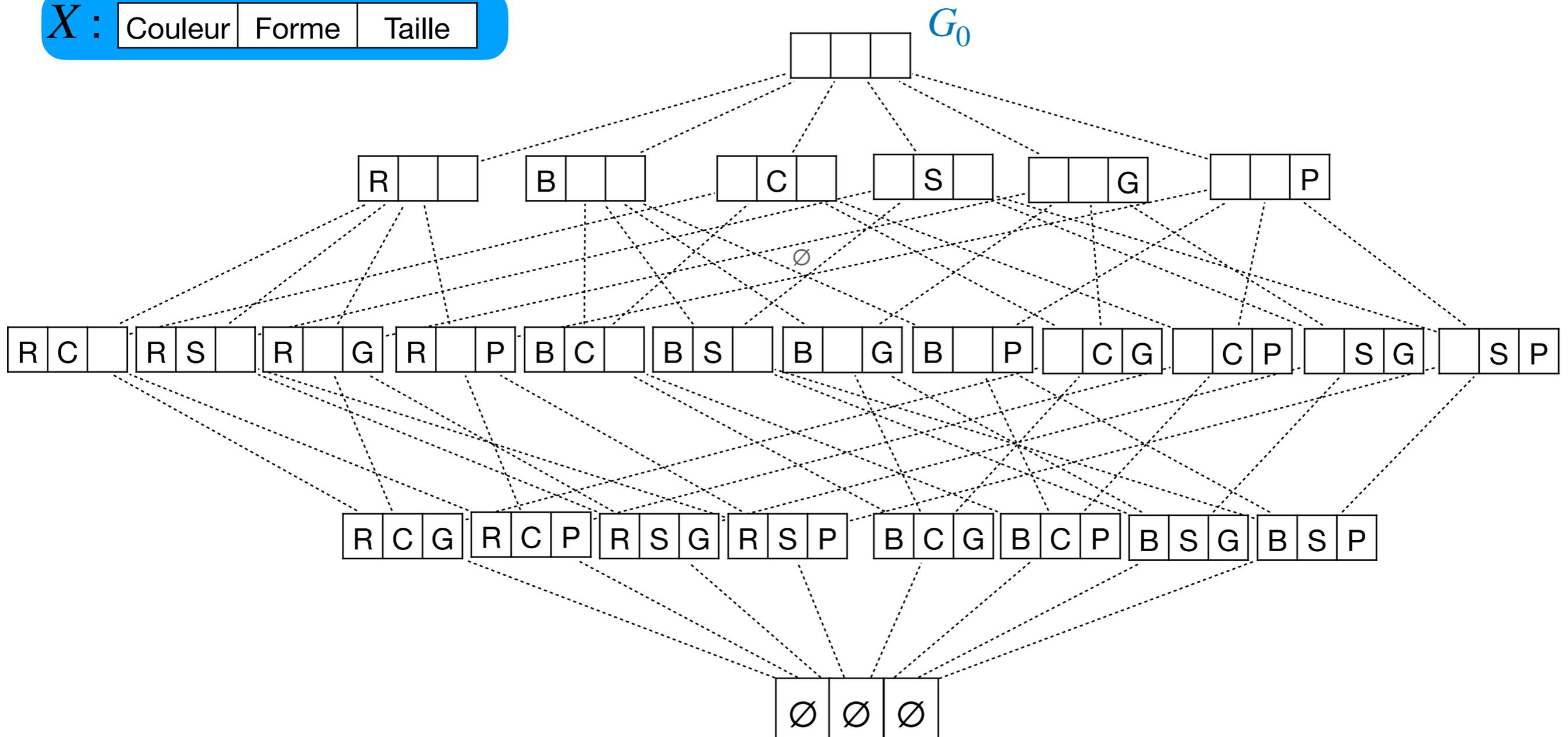
X : Couleur | Forme | Taille



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

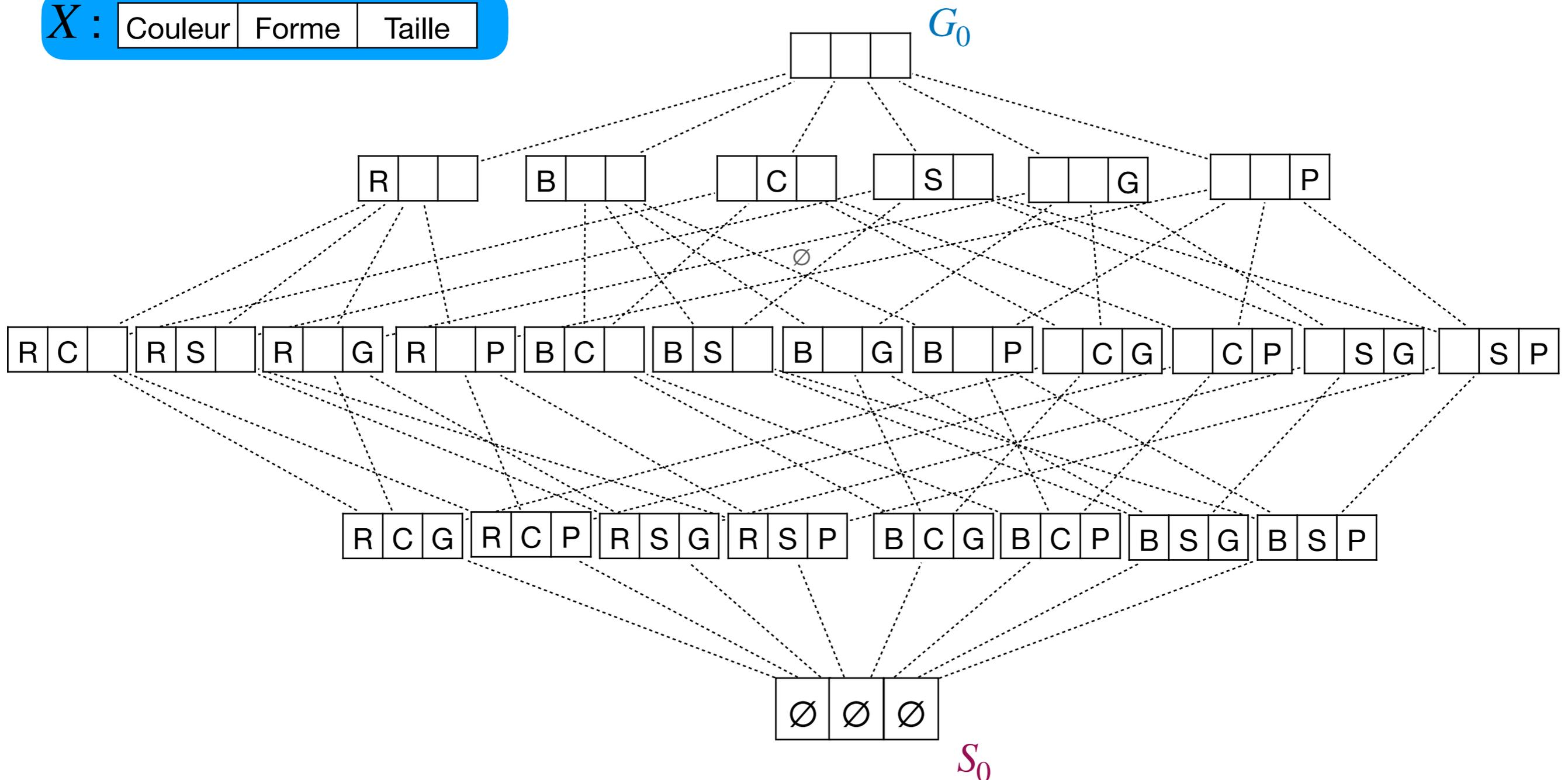
X : Couleur | Forme | Taille



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

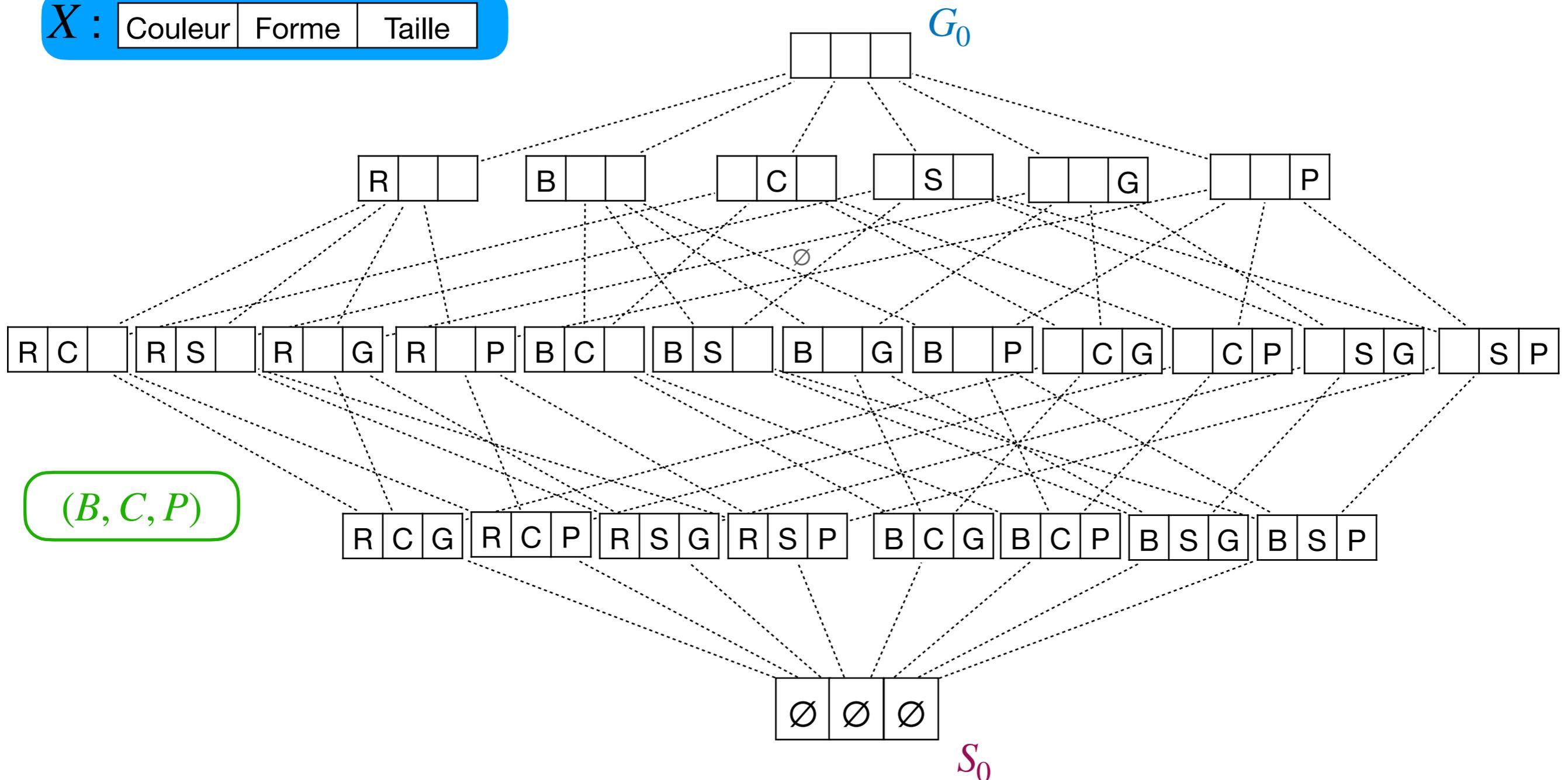
X : Couleur | Forme | Taille



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

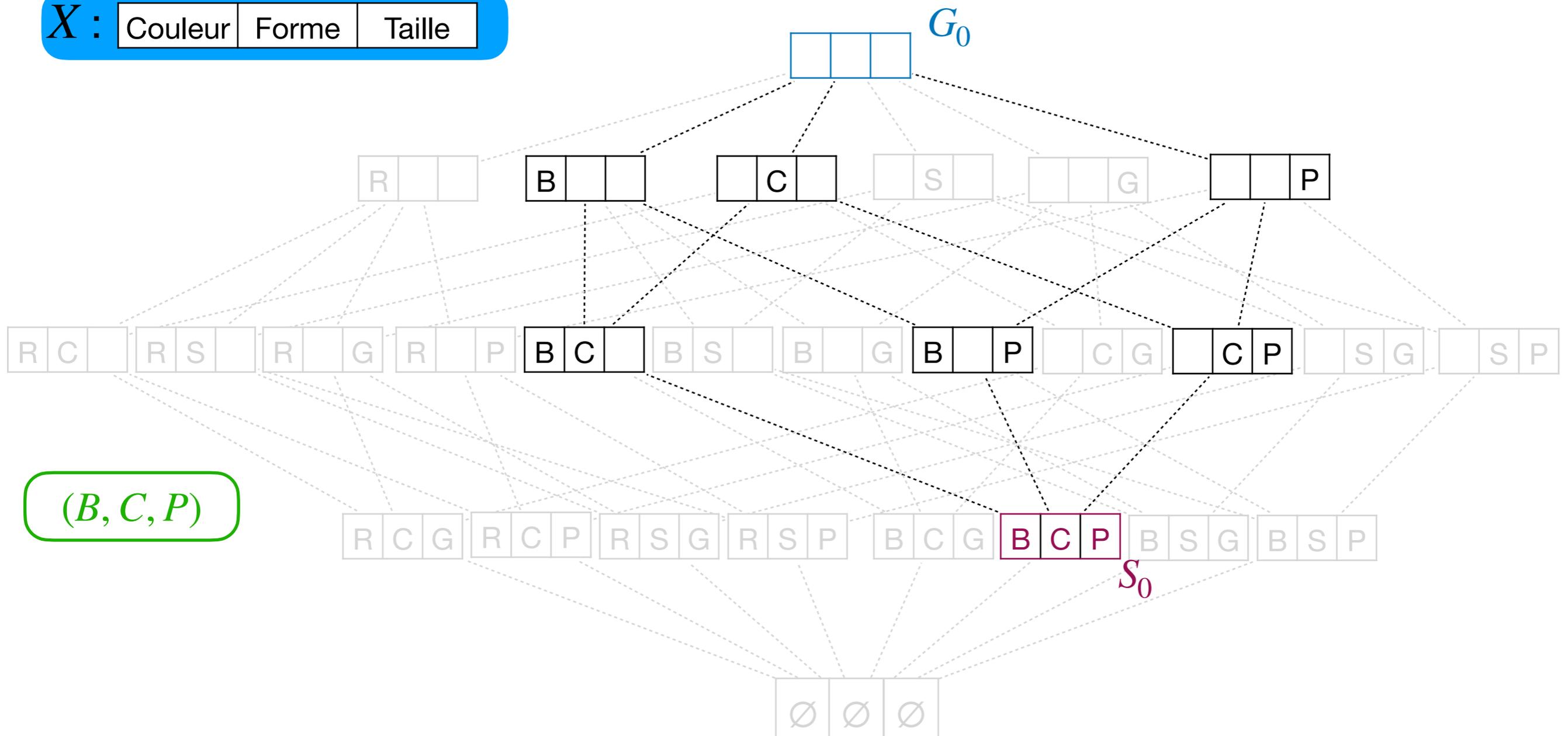
X : Couleur | Forme | Taille



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

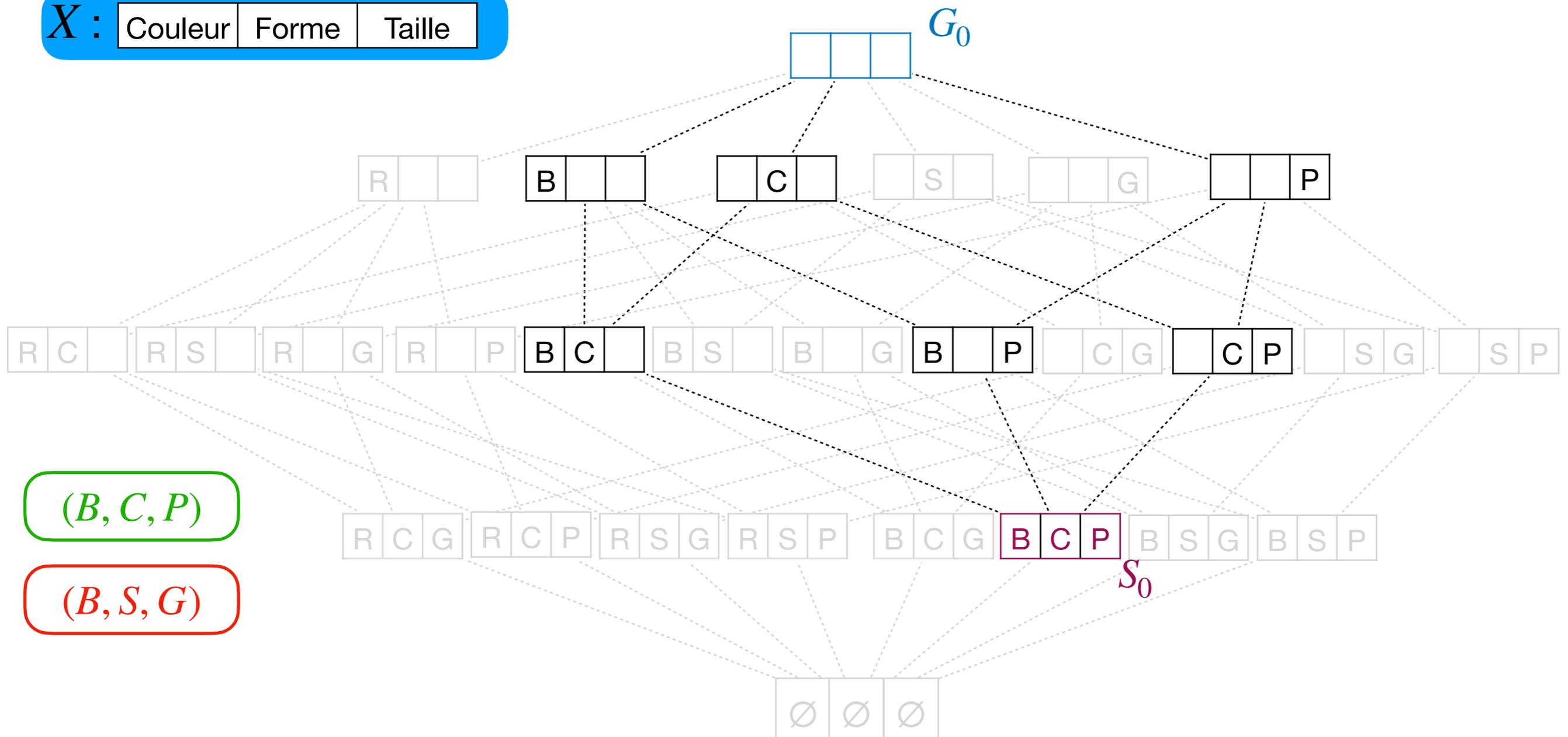
X : Couleur | Forme | Taille



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

X : Couleur | Forme | Taille



Version Space Learning

Algorithme

Algorithme 1 : Algorithme Version Space Learning

Input : Un ensemble d'exemples d'entraînement D avec attributs et classes

Output : Les ensembles S et G définissant l'espace des versions

Initialisation :

$S \leftarrow$ hypothèse la plus spécifique (ex : un ensemble vide ou une instance spécifique)

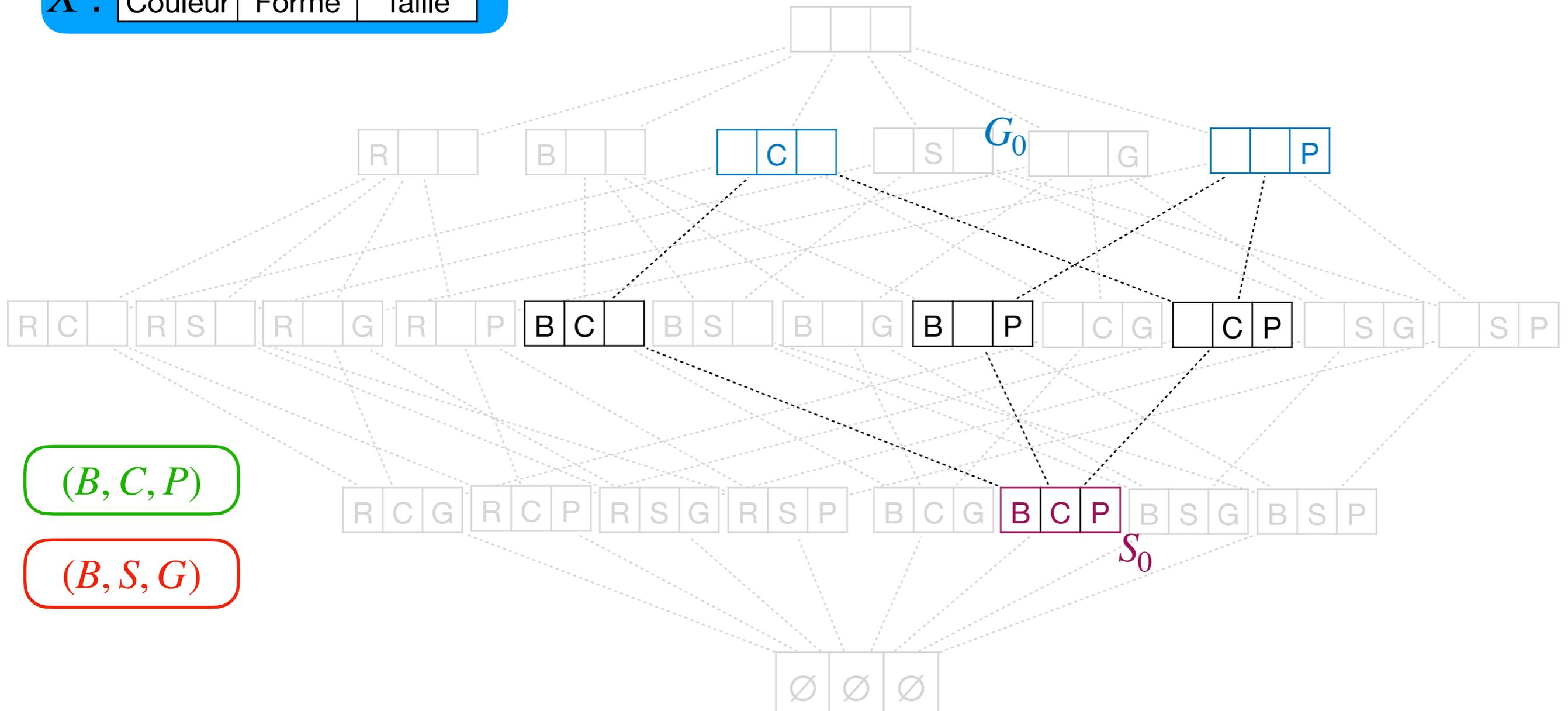
$G \leftarrow$ hypothèse la plus générale (ex : (?, ?, ..., ?))

```
foreach exemple  $(X, y) \in D$  do
  if  $y$  est positif (exemple accepté) then
    foreach  $g \in G$  do
      if  $g$  ne couvre pas  $X$  then
        | Retirer  $g$  de  $G$ 
      end
    end
    foreach  $s \in S$  do
      if  $s$  ne couvre pas  $X$  then
        | Généraliser  $s$  au minimum pour couvrir  $X$ 
      end
    end
    end
    Retirer de  $S$  les hypothèses plus générales que d'autres
  else if  $y$  est négatif (exemple rejeté) then
    foreach  $s \in S$  do
      if  $s$  couvre  $X$  then
        | Retirer  $s$  de  $S$ 
      end
    end
    foreach  $g \in G$  do
      if  $g$  couvre  $X$  then
        | Spécialiser  $g$  pour exclure  $X$ 
      end
    end
    end
    Retirer de  $G$  les hypothèses plus spécifiques que d'autres
  end
end
return  $S, G$ 
```

Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

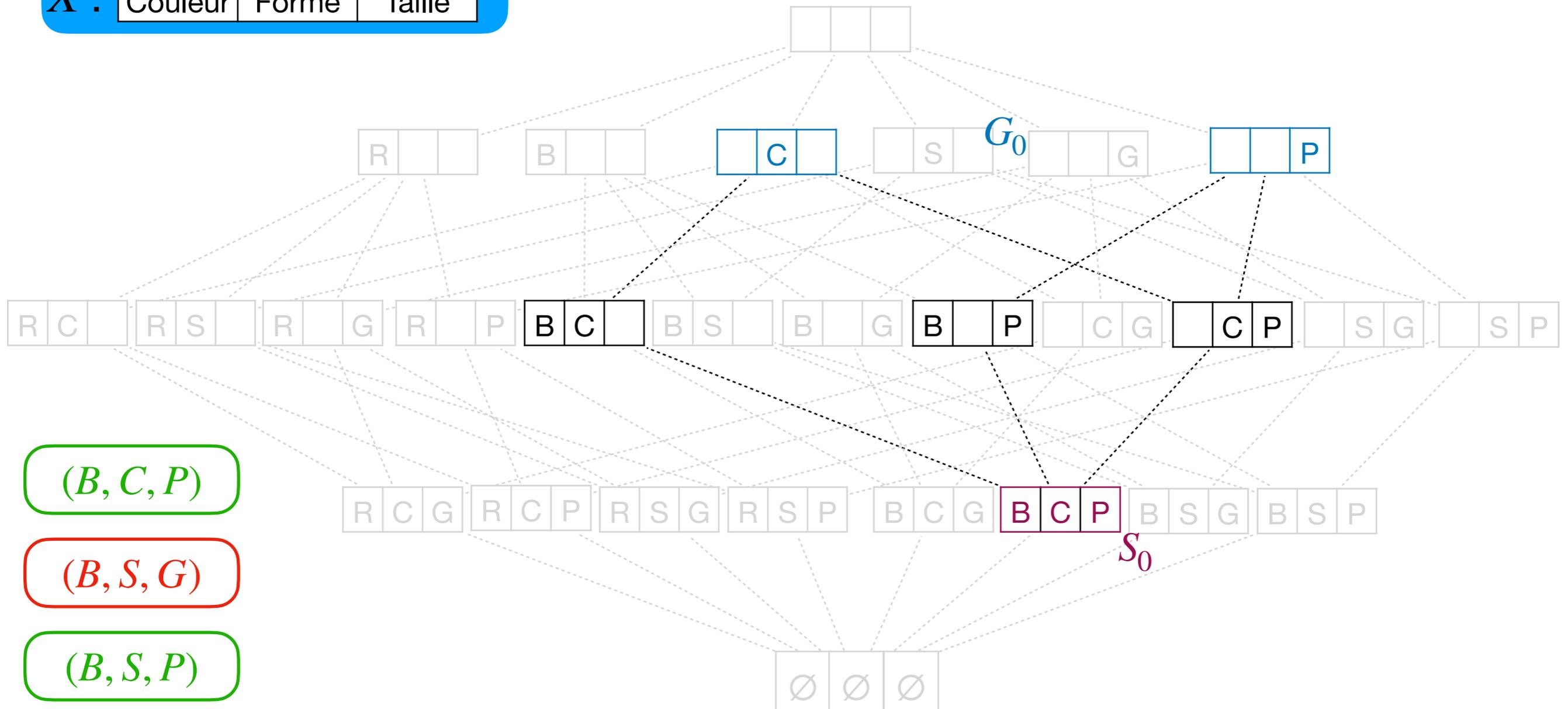
X : Couleur | Forme | Taille



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

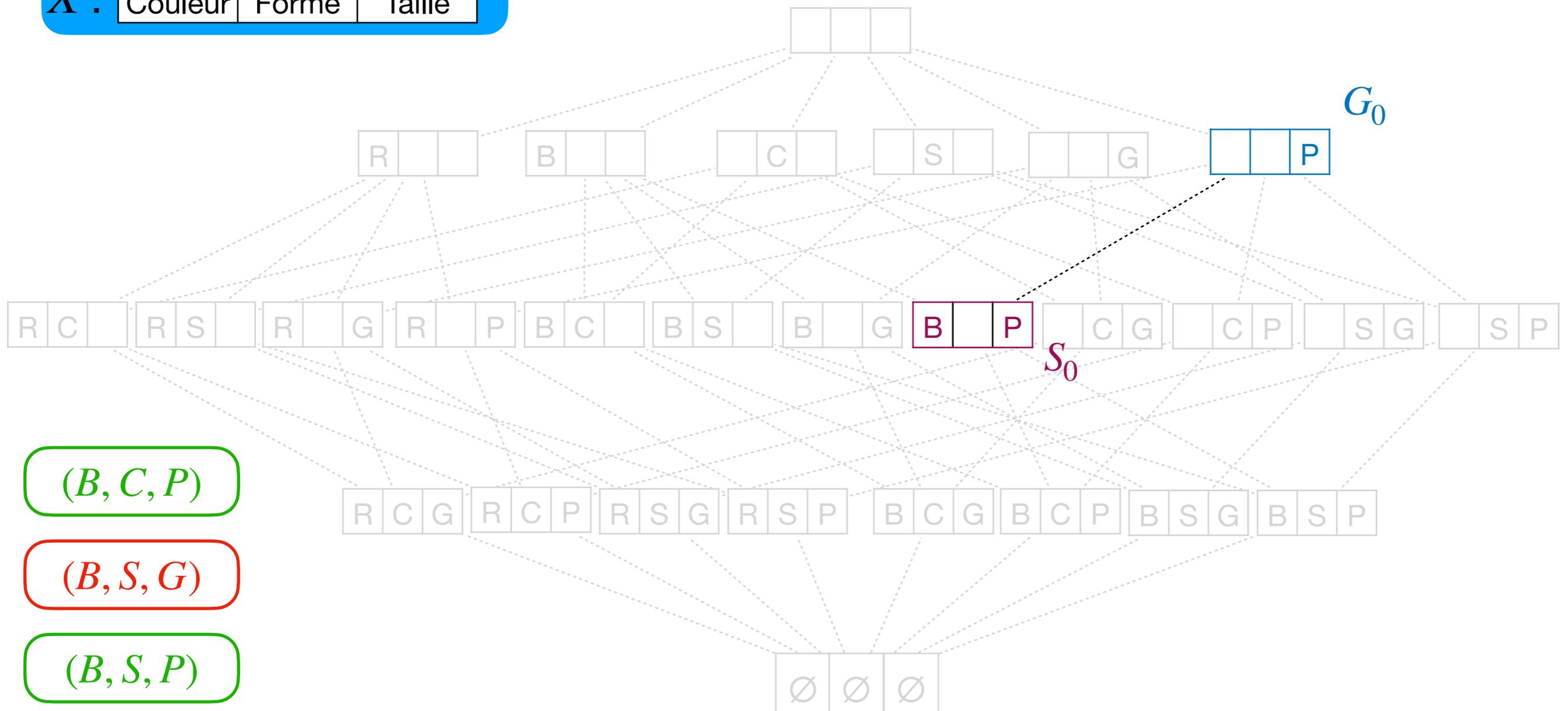
X : Couleur | Forme | Taille



Version Space Learning

Exemple de Treillis (Reconnaissance des formes)

X : Couleur | Forme | Taille



Artificial Intelligence

Cours5 - Version Space Learning

L3 - Informatique

Nadjib Lazaar

Ing - Phd - HDR - Professor - Paris-Saclay University - LISN - LaHDAK

lazaar@lisn.fr

<https://perso.lisn.upsaclay.fr/lazaar/>

24/01/2025