Induction par généralisation

Apprentissage de concepts:

Exemples: E1, E2,... En

p classes disjointes: C1, C2, … Cp

Trouver une fonction caractéristique de chaque classe

Notion de généralité:

<u>Ce morceau</u> de fer chauffé à 500° fond

<u>Cet autre morceau</u> de fer chauffé à 500° fond

<u>Un morceau</u> de fer chauffé à 500° fond

Techniques de généralisation symbolique

Représentation en logique des propositions

- **Abandon d'un littéral**:

si P_i et P_j sont deux propositions (ou conjonction de propositions)

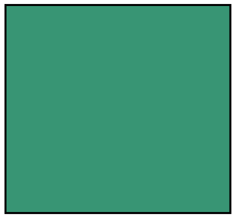
Alors P_i est plus général que P_i & P_j

Notation: $P_i \leq P_i$ & P_j (P_i contient moins d'information que P_i & P_j)

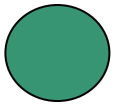
ex. rouge ≤ rouge & carré

Représentation comme sacs de traits

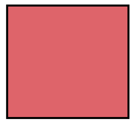
Objet: size {large, small}
form {square, circle}
color {red, green}



size = large
form = square
color = green

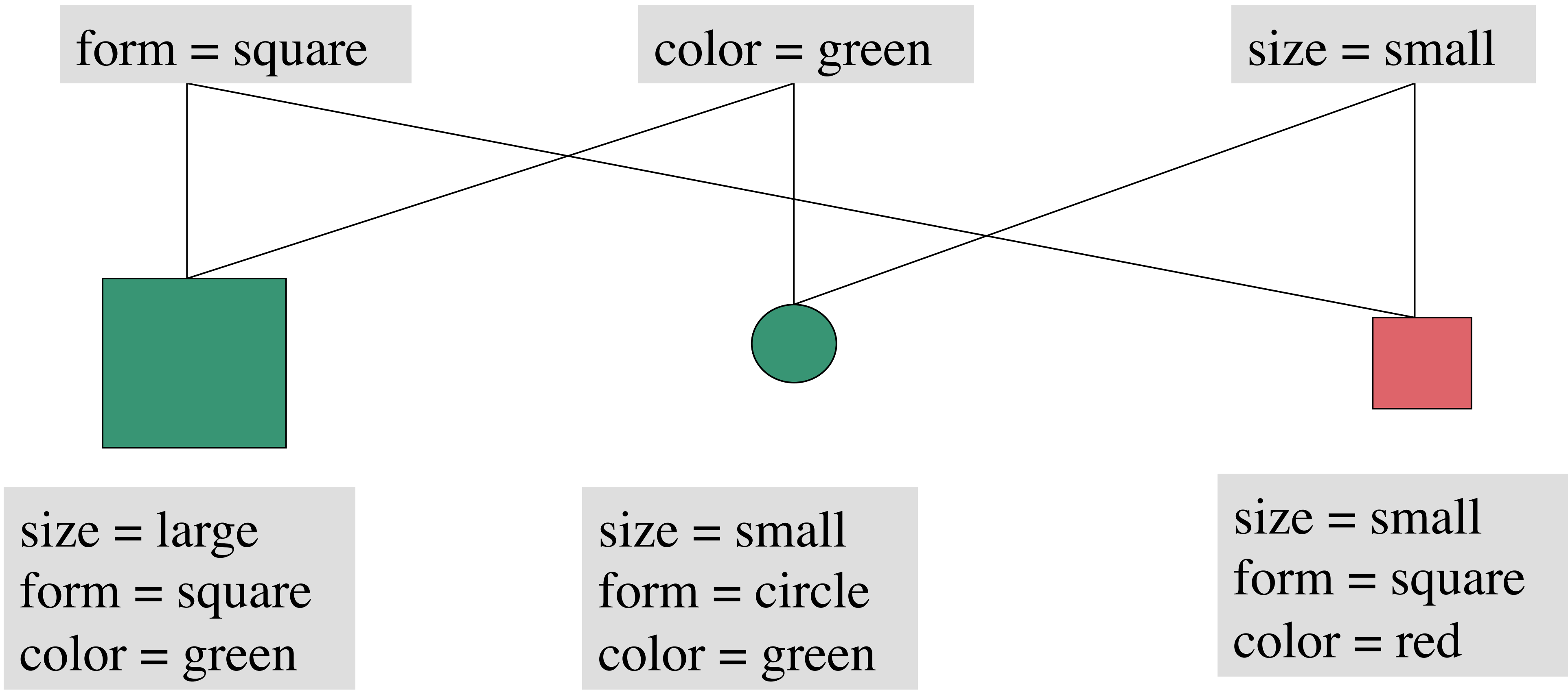


size = small
form = circle
color = green



size = small
form = square
color = red

Généralisation en logique propositionnelle

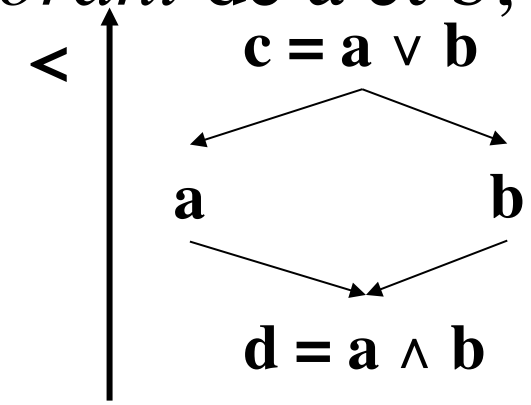


Rappel sur les treillis

- Soit E un ensemble muni d'une relation d'ordre partiel $<$

E est un *treillis* si et seulement si
$\forall (a, b) \in E, \ \exists (c, d) \in E$ tels que:

- c est le *plus petit majorant* de a et b, noté $c = a \vee b$
- d est le *plus grand minorant* de a et b, noté $d = a \wedge b$



Généralisation en logique des propositions

- $A \leq B$ signifie que A est <u>plus général</u> que B
- Nous allons considérer des conjonctions de propositions

Si $d = d1 \wedge d2 \wedge \ldots \wedge dn$ Alors $\forall i \in [1, n]$ $di < d$

Si A est la conjonction des descripteurs d(A), et B la conjonction des descripteurs d(B) alors l'élément $C = A \wedge_g B$ est la conjonction des descripteurs de $d(A) \cap d(B)$

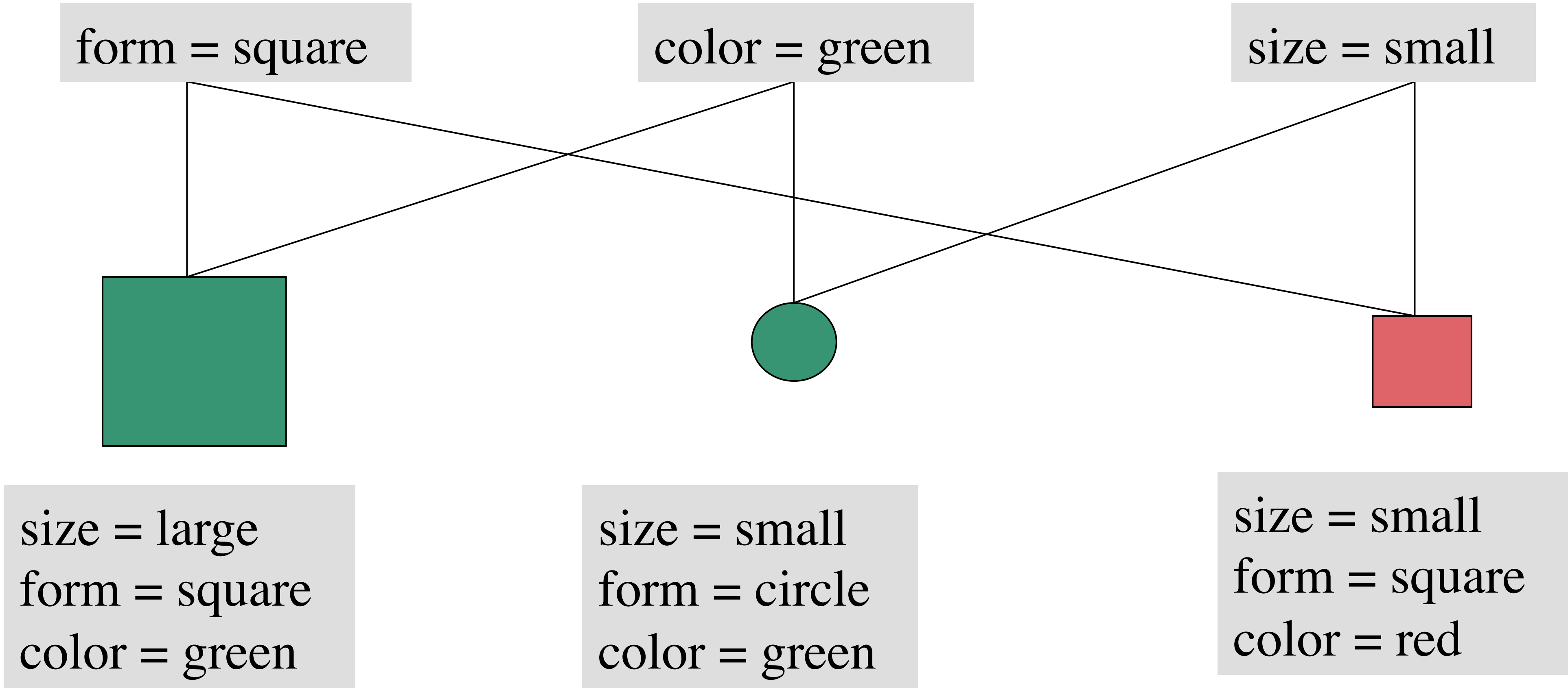
C'est le plus grand généralisé commun à A et B

Généralisation en logique propositionnelle

La description d1 est **plus générale** que d2 (d1 ≤ d2) ssi d1 $\subseteq$ d2

c'est-à-dire ssi d2 $\Rightarrow$ d1 ou d2 $\supseteq$ d1 (sens logique et non ensembliste)

La généralisation de d1 et d2 est d1 $\cap$ d2



Treillis des descripteurs

- L'ensemble $\mathcal{D}$(D) des conjonctions de descriptions atomiques D possède aussi une structure de treillis

$\forall$ (X, Y) $\in \mathcal{D}$(D)

$\exists$U $\in \mathcal{D}$(D) tel que U est la description la plus général qui contient les descriptions X et Y:
$U = \underline{X \vee_g Y} = X \wedge Y$

$\exists$V $\in \mathcal{D}$(D) tel que V est la description la plus spécifique qui est plus générale que X et Y:
$V = gen(X, , Y) = \underline{X \wedge_g Y} = X \cap Y$

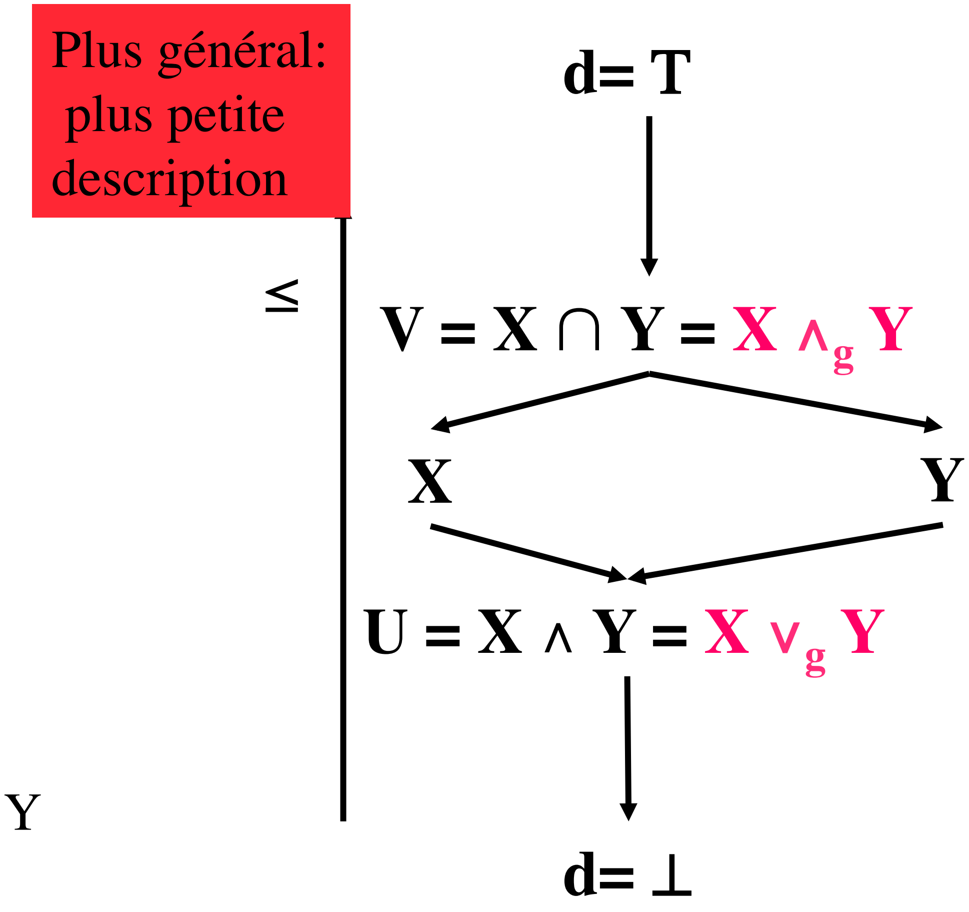
Treillis des descripteurs

- L'ensemble $\mathcal{D}$(D) des conjonctions de descriptions atomiques D possède aussi une structure de treillis

$\forall$ (X, Y) $\in$ D(D)

$\exists$U $\in$ D(D) tel que U est la description la plus général qui contient les descriptions X et Y: U = $X \vee_g Y$ = X $\wedge$Y

$\exists$V $\in$ D(D) tel que V est la description la plus spécifique (la plus grande) qui est plus générale que X et Y:
V = gen(X, , Y) = $X \wedge_g Y$ = X$\cap$Y

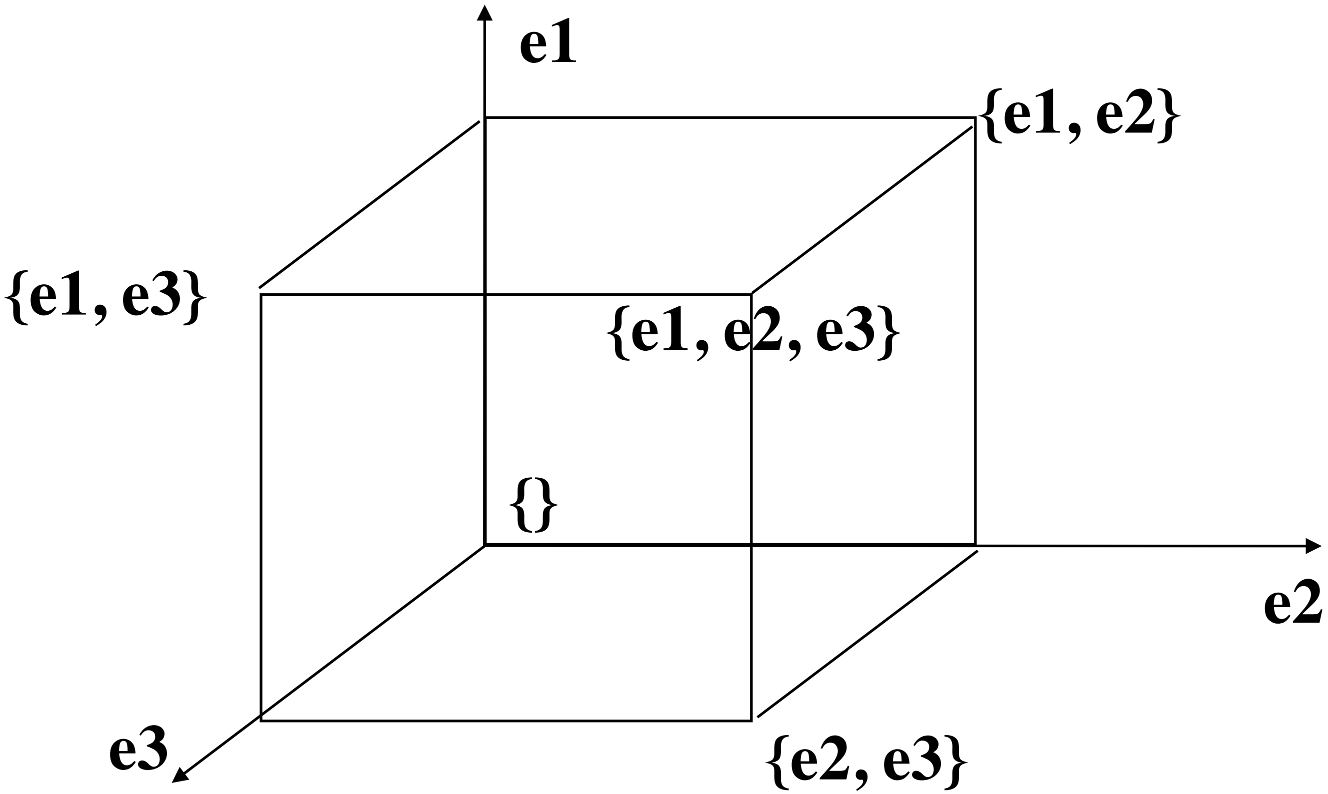


Treillis des exemples

e1 = (couleur=rouge) & (taille = 3) & (forme = cercle)

e2 = (couleur=rouge) & (taille = 5) & (forme = triangle)

e3 = (couleur=bleu) & (taille = 6) & (forme = rectangle)

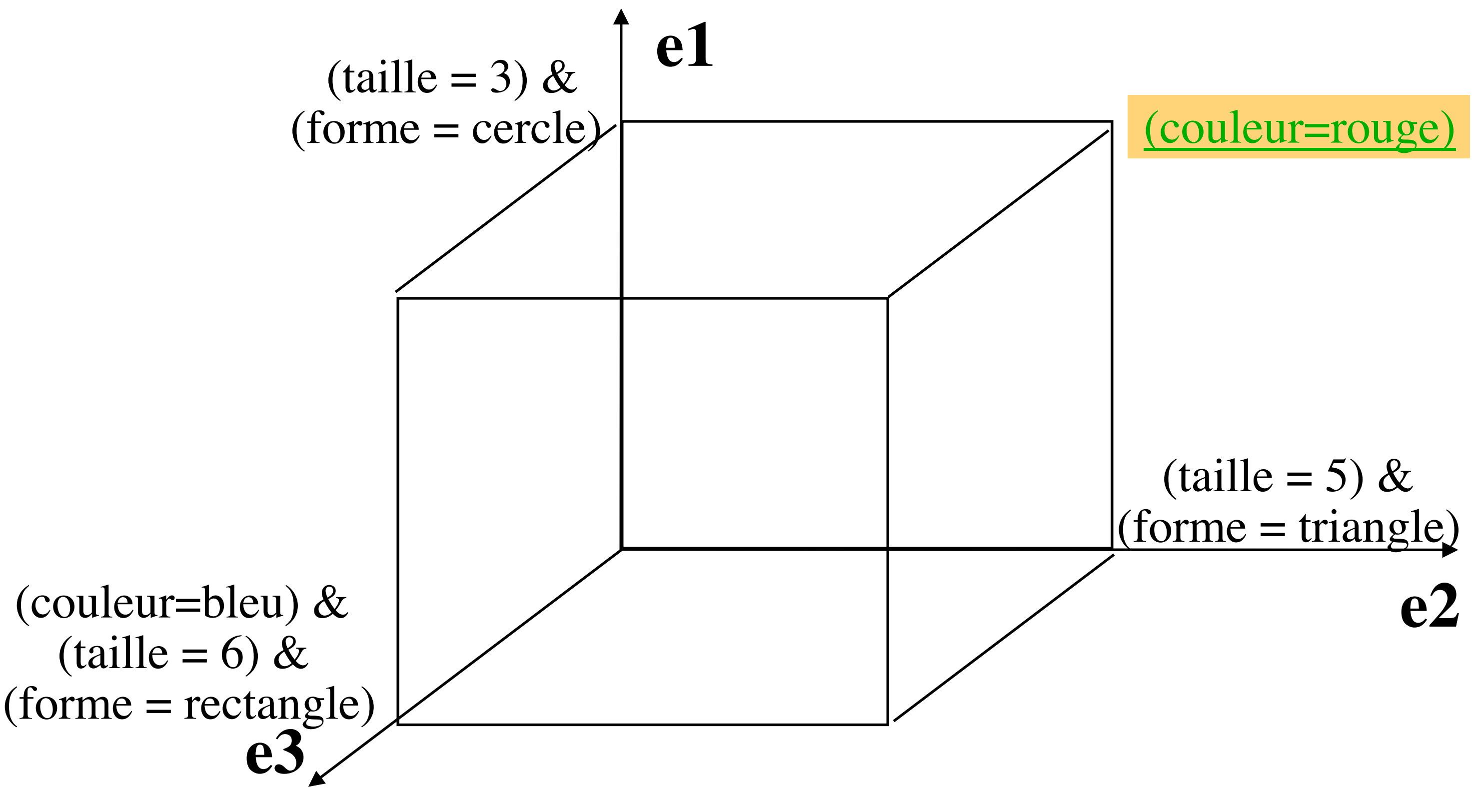


Treillis des exemples: pauvreté de la généralisation

e1 = (couleur=rouge) & (taille = 3) & (forme = cercle)

e2 = (couleur=rouge) & (taille = 5) & (forme = triangle)

e3 = (couleur=bleu) & (taille = 6) & (forme = rectangle)



Utilisation de connaissances

- **Extension des frontières** dans le cas d 'attributs valués sur des domaines ordonnés:

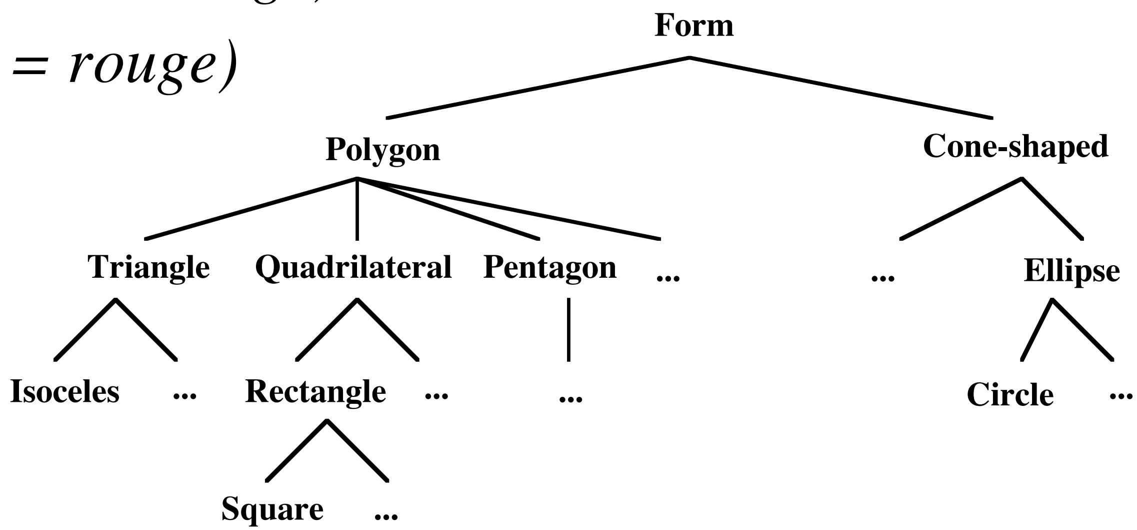
(taille ≥ 7) < (taille ≥ 37)

- **Remontée dans une hiérarchie conceptuelle**:

polygone < carré

(forme = polygone) & (couleur = rouge) <

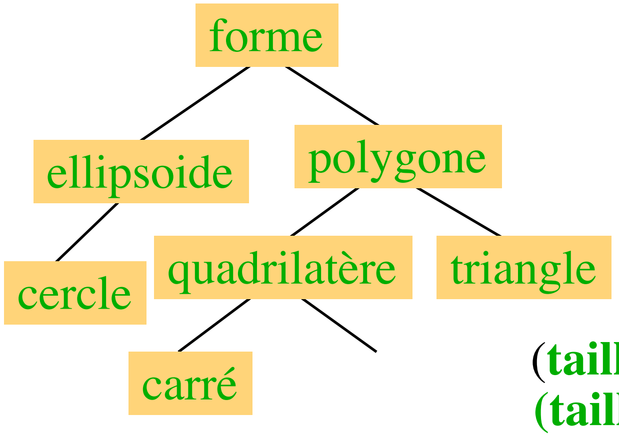
(forme = carré) & (couleur = rouge)

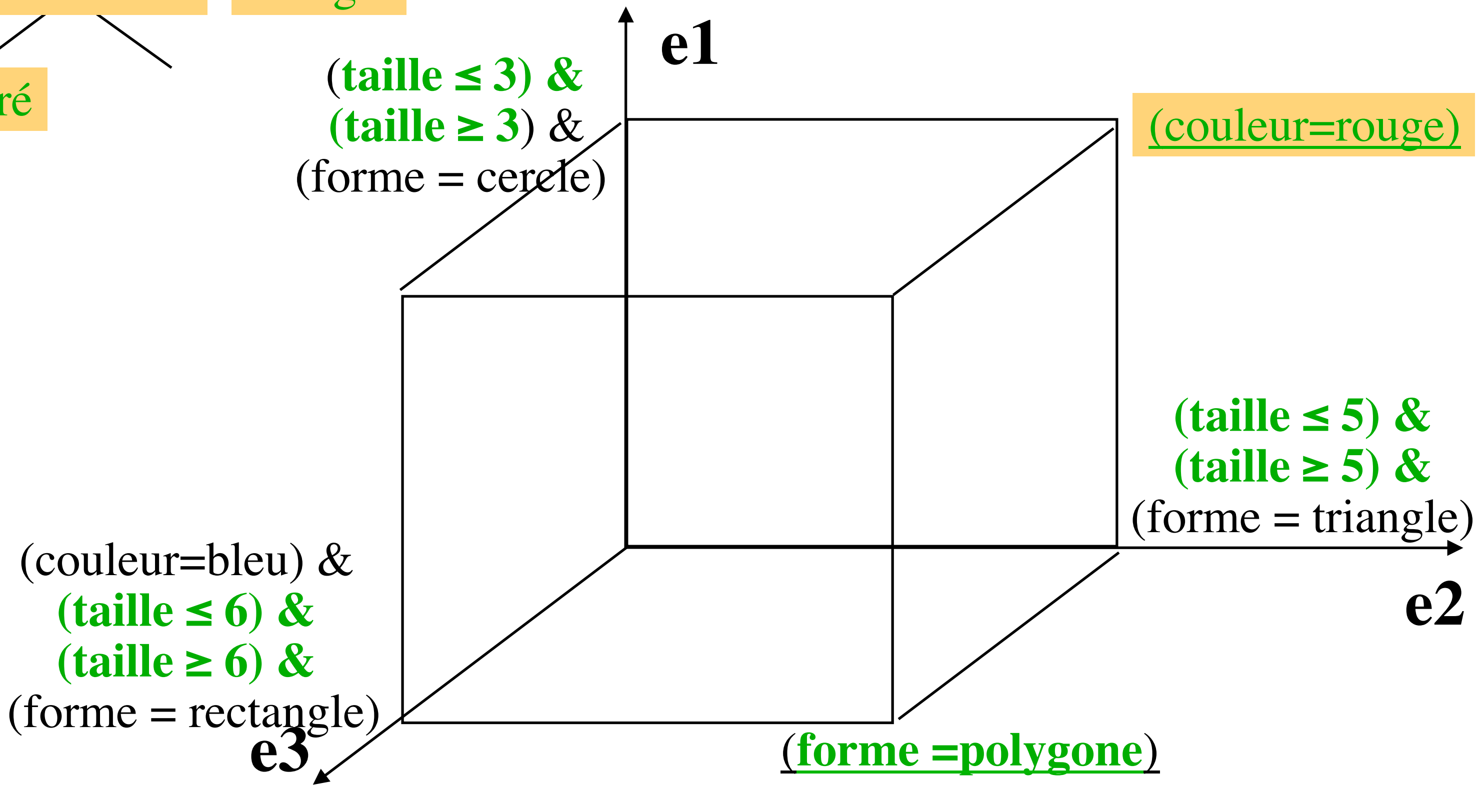


Treillis des exemples: Introduction de connaissances

$X \geq Y$ & $Y \geq Z \Rightarrow X \geq Z$

$X \leq Y$ & $Y \leq Z \Rightarrow X \leq Z$

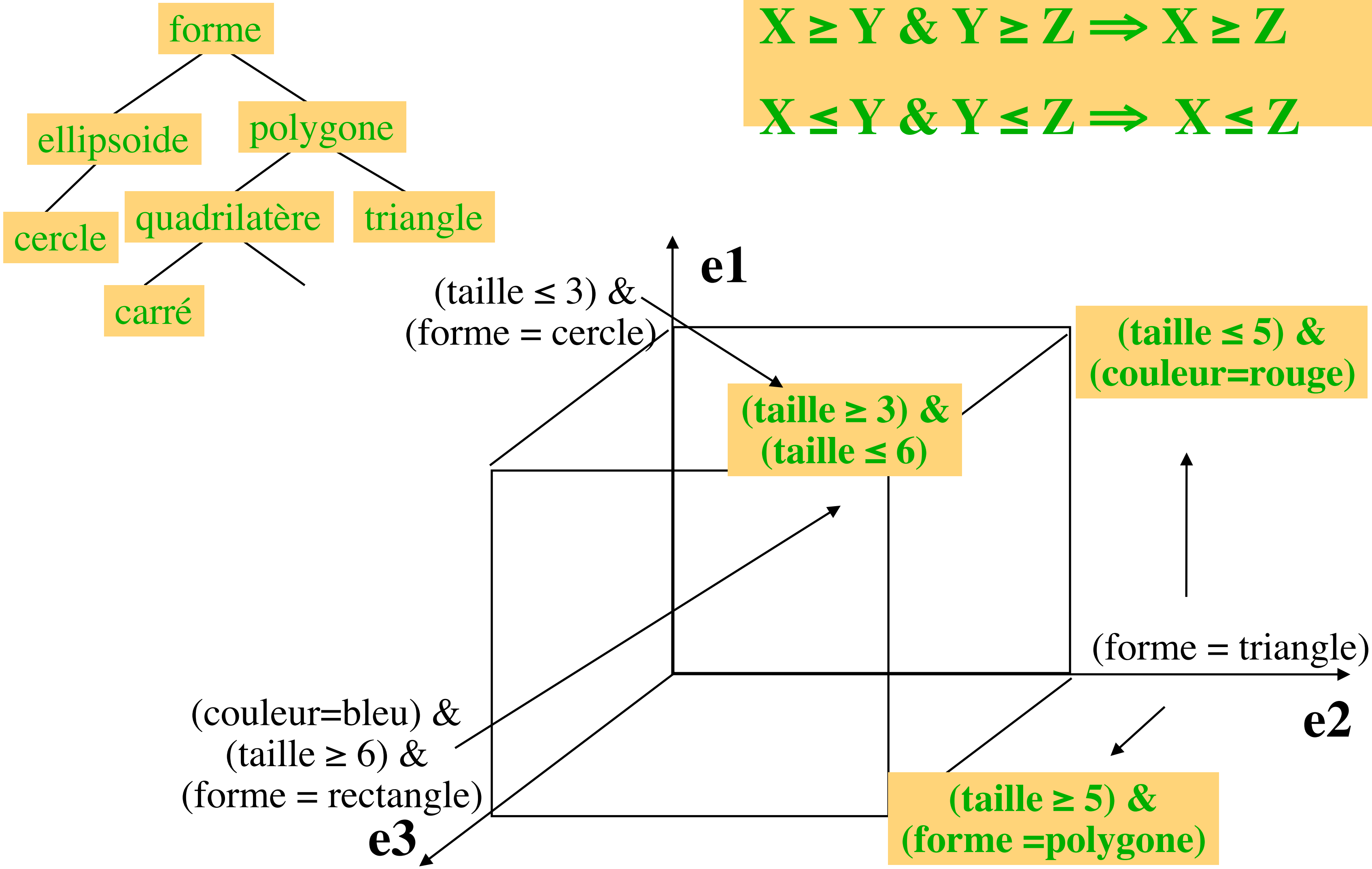




Treillis des exemples: connaissances

$X \geq Y \;\&\; Y \geq Z \Rightarrow X \geq Z$

$X \leq Y \;\&\; Y \leq Z \Rightarrow X \leq Z$



Recherche de la bonne généralisation *(Mitchell, 81)*

Étant donné:

- Un langage de représentation des instances I
- Un langage de représentation des généralisations G
- Un prédicat de couverture couv(G, I)
- Un ensemble d'exemples positifs {Ei} et
- Un ensemble d'exemples négatifs {CEj}

Trouver: une généralisation G correcte et cohérente

G correcte: pour tout Ei, couv(G, Ei)

G cohérente: pour tout CEj, ¬couv(G, CEj)

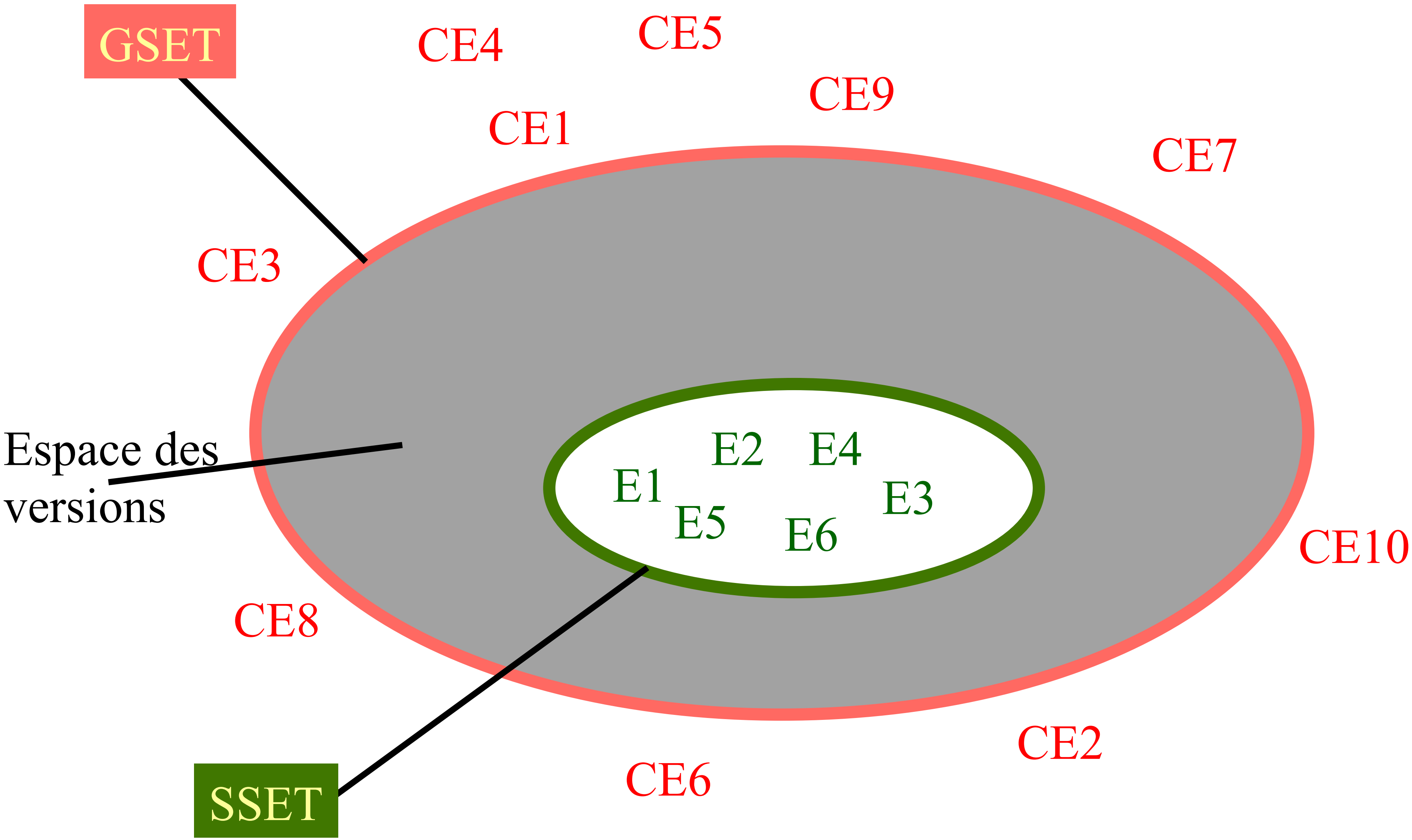
Espace des versions

1. Trouver la bonne généralisation se ramène à une RECHERCHE dans un espace de description

2. Algorithme de généralisation « incrémental » - donner exemples 1 à 1

3. **Relation d'ordre**: généralité, couverture

Espace des versions



Bornes de l'espace de recherche

Sset = {s | s est une généralisation *maximalement spécifique*}

s est *maximalement spécifique* si s est cohérente et s'il n'existe pas de généralisation s' plus spécifique que s

Gset = {g | g est une généralisation *maximalement générale*}

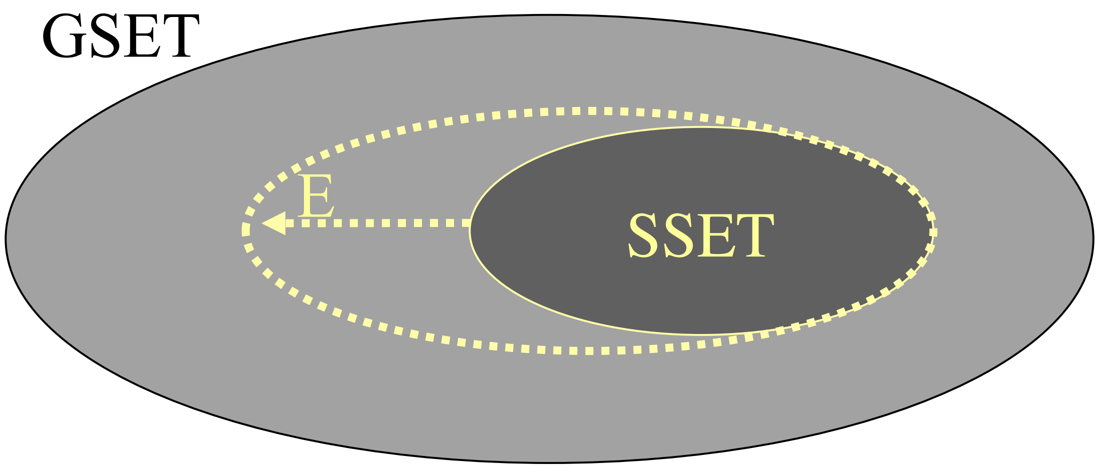
g est *maximalement générale* si g est cohérente et s'il n'existe pas de généralisation g' plus générale que g

Recherche ascendante en largeur

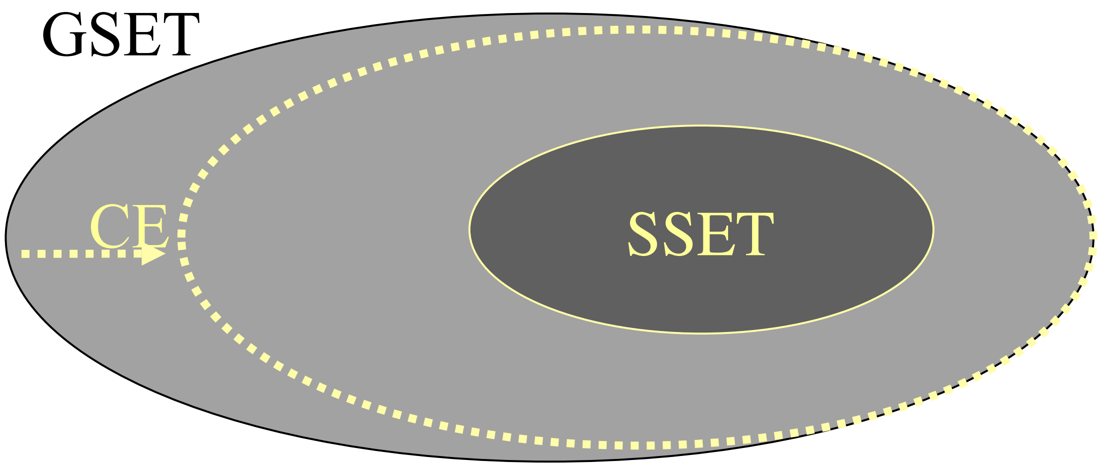
- Les révisions de S à chaque nouvel exemple sont consistantes avec les exemples positifs passés
- On ne stocke pas les exemples positifs
- On stocke les exemples négatifs pour vérifier qu ’ il n ’ y a pas de surgénéralisation

Algorithme d ’ élimination des candidats: principe

Ajout d ’ un exemple E dans l’ espace des versions



Ajout d ’ un contre-exemple CE dans l’ espace des versions



Traitement de S

Pour chaque instance i

Si i est négatif (contre-exemple),

- on ne retient que les généralisations de S qui ne couvrent pas i

Sinon,

- Généraliser les éléments de S afin qu'ils couvrent i
- Enlever de S les éléments qui couvrent des exemples négatifs

Traitement de G

Pour chaque instance i

Si i est positif (exemple),

- on ne retient que les généralisations de G qui couvrent i

Sinon,

- Spécialiser les éléments de G afin qu'ils ne couvrent plus i
- Enlever de G les éléments qui ne sont pas plus généraux qu'au moins un exemple de S

Algorithme d'élimination des candidats

- Initialiser S et G par les ensembles de généralisation les plus spécifiques et les plus générales avec le premier exemple positif

- Pour chaque exemple suivant (ou contre-exemple)
  - Traiter S
  - Traiter G

Jusqu'à convergence (ou jusqu'à G-S=Ø)

Les propriétés de l'algorithme

- **S résume les informations relatives aux exemples positifs** (il ne sont pas stockés explicitement)
- **G résume les informations relatives aux exemples négatifs** (il ne sont pas stockés explicitement)
- Le résultat est **indépendant de l'ordre** de présentation
- Les exemples **positifs** déplacent **S**, les **négatifs, G**
- Si S et G deviennent vide, il n'y a pas d'hypothèse cohérente dans le langage.

Exemple (énoncé)

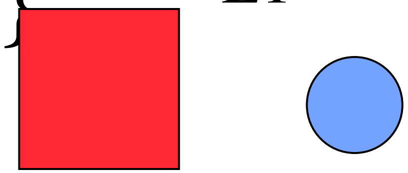
Objet: taille {grand, petit}

forme_géométrique {carré, cercle}

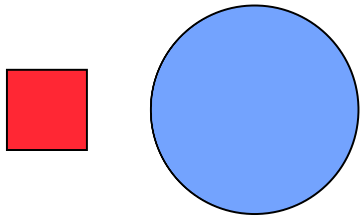
couleur {rouge, bleu}

S = Ø, G = {(?,?,?)}

E1



E2

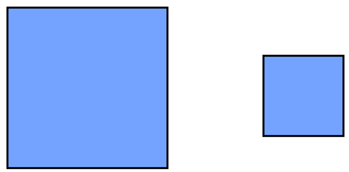


E1 (+) : (grand carré rouge) & (petit cercle bleu)

E2 (+) : (petit carré rouge) & (grand cercle bleu)

E3 (-) : (grand carré bleu) & (petit carré bleu)

E3

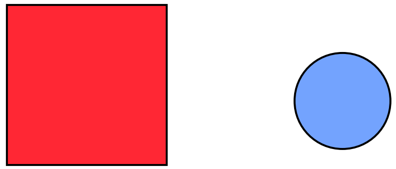


Exemple (suite)

E1 (+) : (grand carré rouge) & (petit cercle bleu)

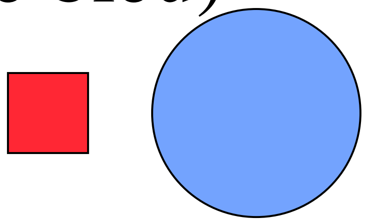
S1 = {(grand carré rouge) (petit cercle bleu)}

G1 = {(?,?,?)(?,?,?)}



E2 (+) : (petit carré rouge) & {grand cercle bleu)

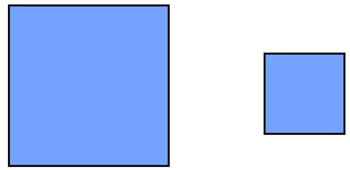
S2 = {(? carré rouge) (? cercle bleu)}



G2 = {(?,?,?)(?,?,?)}

Exemple (suite)

E3 (-) : (grand carré bleu) & (petit carré bleu)



S3 = {(? carré rouge) (? cercle bleu)}

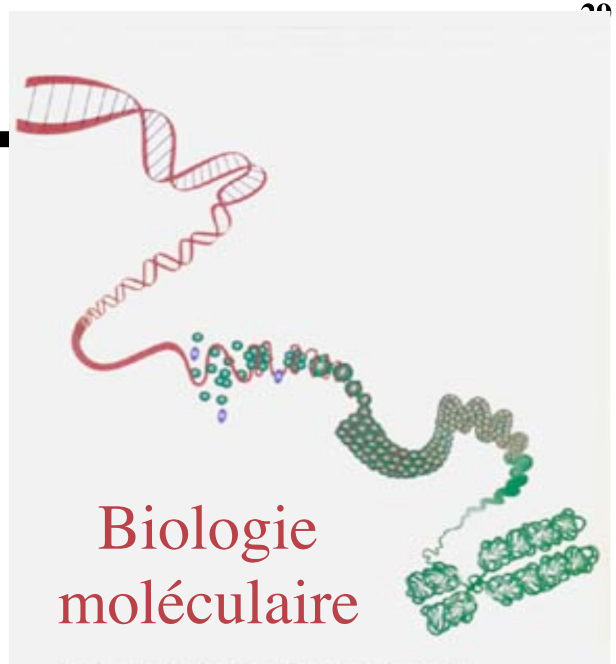
G3 = {(?,?,rouge)(?,?,?), (?,?,?)(?, cercle,?)}

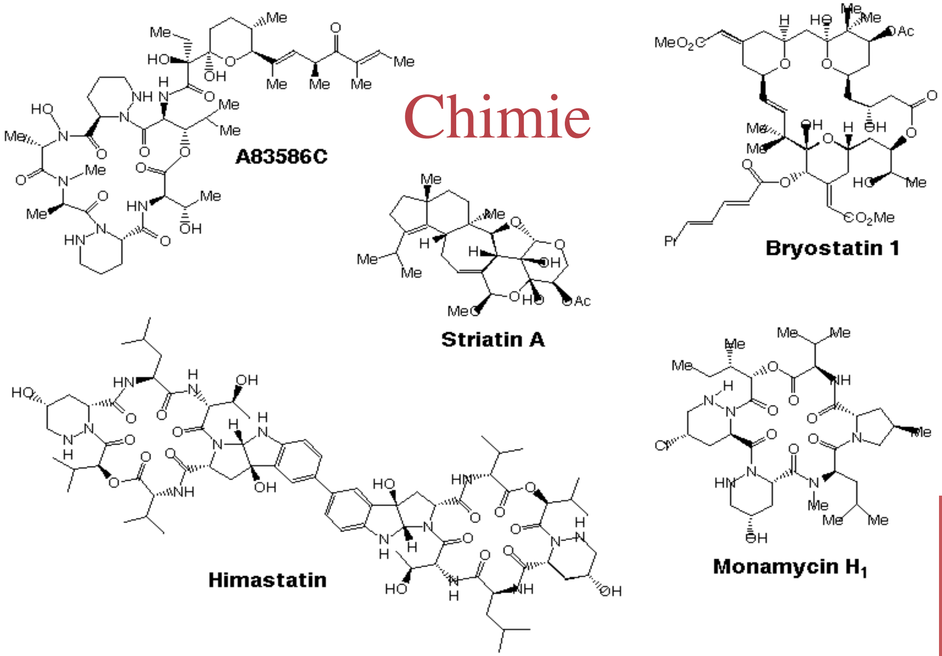
Exercice: calcul S et G

Cas 1: convergence
- E1 (+) (grand, rouge, cercle)
- E2 (-) (petit, rouge, carré)
- E3 (+) (petit, rouge, cercle)
- E4 (-) (grand, bleu, cercle)

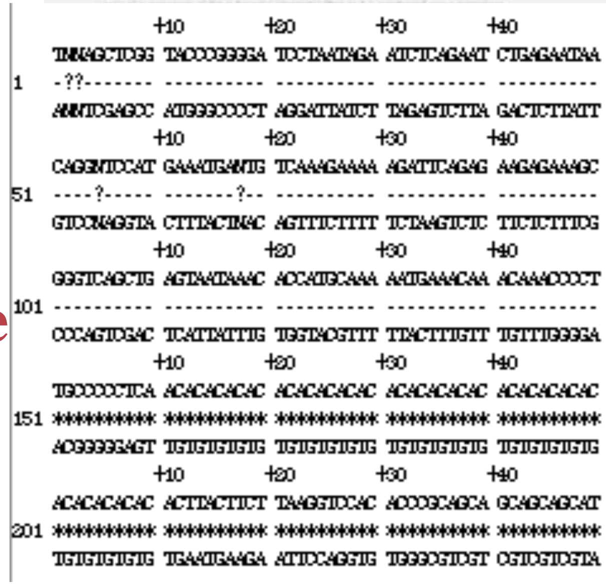
Cas 2: pas de convergence
- E1 (+) (grand, rouge, cercle)
- E2 (-) (petit, bleu, carré)
- E3 (+) (petit, rouge, cercle)
- E4 (-) (moyen, vert, triangle)

Cas 3: langage insuffisant
- E1 (+) (grand, rouge, cercle)
- E2 (-) (grand, bleu, cercle)
- E3 (+) (petit, bleu, cercle)





Insuffisance de l'approche propositionnelle

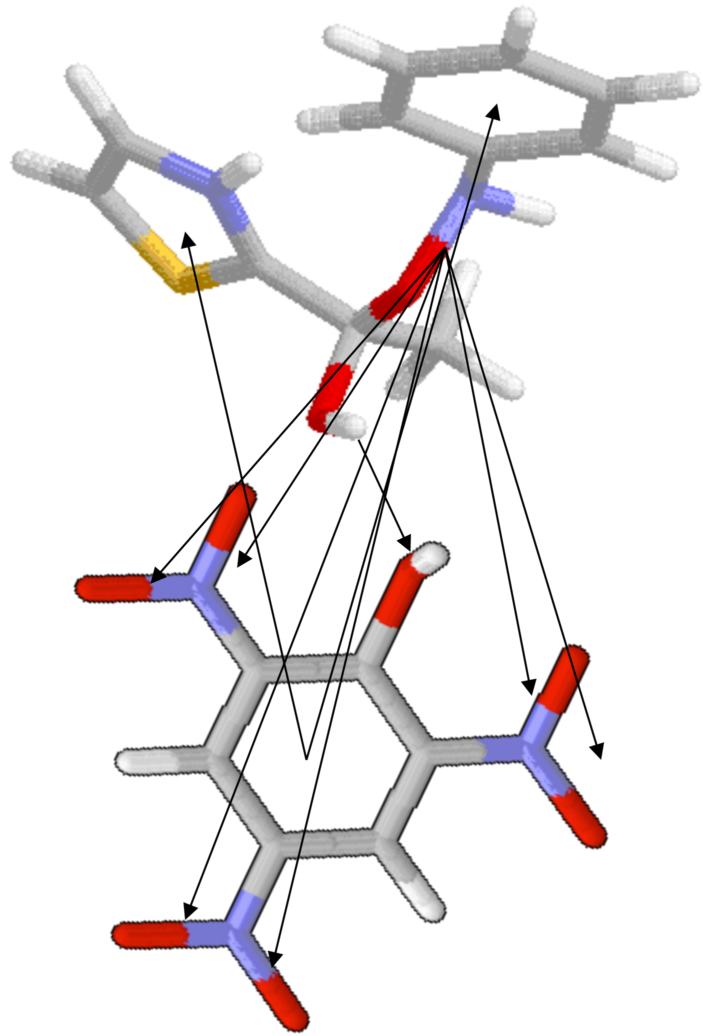


Induction structurelle

Induction de structures et appariement

- L'intelligence artificielle ne traite pas uniquement des **sacs de traits**, mais aussi des **structures**

- Rôle de l'**appariement** dans l'induction structurelle



Cadre formel

- Induction structurelle
  - Interference Matching, 1978, Hayes-Roth & McDermott,
  - THOT, 1979, Vere,
  - AGAPE, 1983, Kodratoff & Ganascia
- Généralisation inductive
  - Plotkin 1971
- Inductive logic programming (PLI en français)
  - Muggleton 1992
  - …

Représentation en logique du 1er ordre

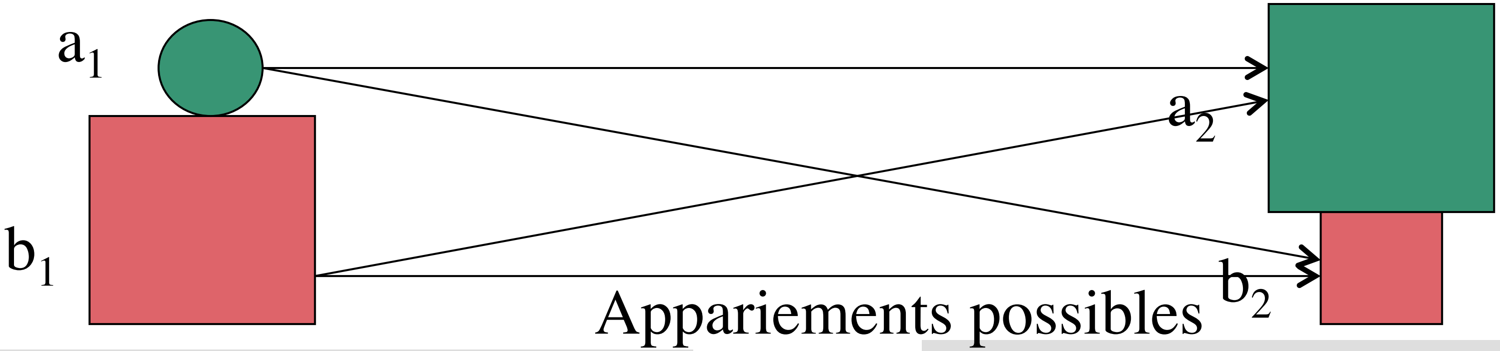
Avec des **structures**, la détection des propriétés requiert d'**apparier des sous-structures**

- La description d_1 est **plus générale** que d_2 ($d_1 \leq d_2$)

ssi $\underline{\exists\sigma}$ tel que $d_1\underline{\sigma} \subset d_2$ *(ex square(?x) ≤ square(b_1) ∧ red(b_1))*

- **Généralisations** de d_1 et d_2: tous les d tels que

$\exists\sigma_1$ avec $d\sigma_1 \subset d_1$ et $\exists\sigma_2$ avec $d\sigma_2 \subset d_2$



circle(a_1) ∧ small(a_1) ∧ green(a_1) ∧ square(b_1) ∧ large(b_1) ∧ red(b_1) ∧ on(a_1, b_1)

square(a_2) ∧ large(a_2) ∧ green(a_2) ∧ square(b_2) ∧ small(b_2) ∧ red(b_2) ∧ on(a_2, b_2)

Généralisations avec structures: rôle de l'appariement

$green(?x) \wedge square(?y) \wedge red(?y) \wedge on(?x, ?y)$

$a_1 \leftrightarrow a_2$, $b_1 \leftrightarrow b_2$

$small(?x) \wedge square(?y) \wedge large(?y)$

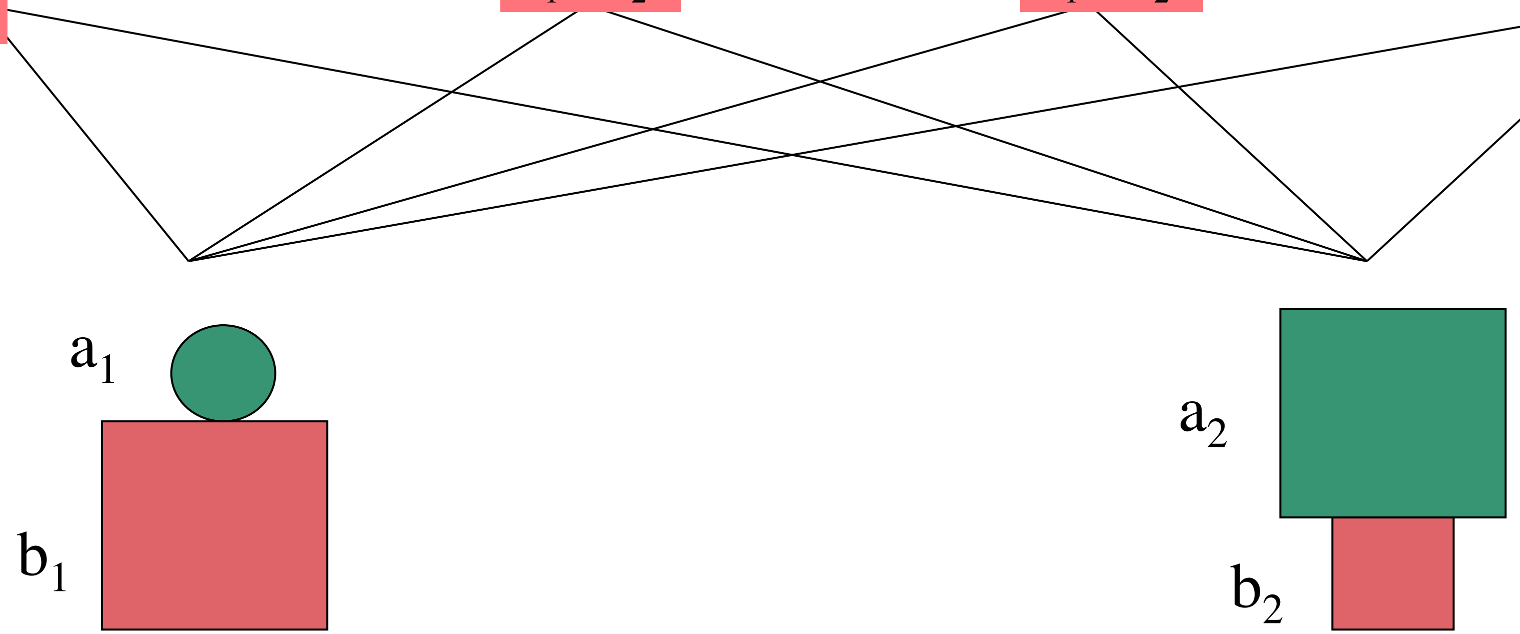
$a_1 \leftrightarrow b_2$, $b_1 \leftrightarrow a_2$

$small(?x) \wedge square(?y) \wedge red(?y) \wedge$

$a_1 \leftrightarrow b_2$, $b_1 \leftrightarrow b_2$

$green(?x) \wedge square(?y) \wedge large(?y) \wedge$

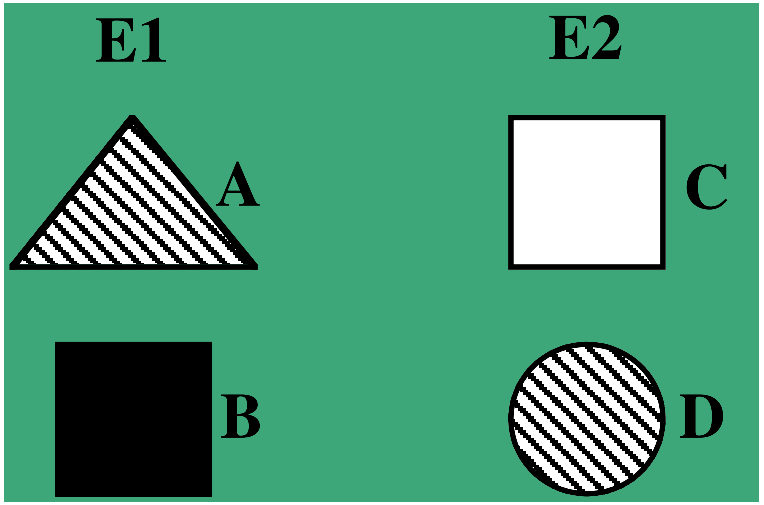
$a_1 \leftrightarrow a_2$, $b_1 \leftrightarrow a_2$



$circle(a_1) \wedge small(a_1) \wedge green(a_1) \wedge square(b_1) \wedge large(b_1) \wedge red(b_1) \wedge on(a_1, b_1)$

$square(a_2) \wedge large(a_2) \wedge green(a_2) \wedge square(b_2) \wedge small(b_2) \wedge red(b_2) \wedge on(a_2, b_2)$

Généralisation objets structurés: questions



- Il y a deux objets l'un sur l'autre

- L'objet du dessus est un polygone

- Il y a un objet rayé

- Il y a un carré plein

- etc.

Connaissances: introduction du nombre *deux*, des notions de *polygone* et de *plein*

Préconception: le nombre de généralisés potentiels est si important qu 'il faut introduire des *contraintes syntaxiques* et *sémantique* qui orientent la généralisation

L’espace des généralisés

- **Logique des propositions: trellis**

Un généralisé minimal pour tout couple d’objets

Apprentissage *règles d’association*

→ *Analyse Formelle de Concepts*

- **Logique des précicats: cylindres**

Un grand nombre de généralisation pour un couple d’objets...

Apprentissage inductif

→ *Programmation Logique Inductive*