Programme

- Introduction
- **Le langage de la LP (syntaxe)**
- La sémantique de la LP
- Équivalence logique et Substitution
- Conséquence logique
- Formes normales et clausale
- Méthode de résolution
- Méthode des tableaux
- Méthode de Davis et Putnam
- Initiation à la logique des prédicats

Syntaxe de la logique des propositions

- Soit l'ensemble des mots définis sur l'alphabet $S \cup C \cup D \cup L$ où :
  - **S** est un ensemble dénombrable de symboles propositionnels

    Notés en lettre minuscule dans les exemples S={p,q,r…}
  - C={¬,∧,∨,→,↔} est l'ensemble des connecteurs logiques

    non (négation), et (conjonction), ou (disjonction), implique /si-alors (implication), équivalent/si-et-seulement-si (équivalence)
  - D={(,)} est un jeu de parenthèses
  - L={T,⊥} les constantes logiques

    Top(True/Vrai), Bottom(Absurde)

FBF : Formules Bien Formées

- On définit **PROP(S)**, l'ensemble des fbf de la logique des propositions (ou propositions), construites sur S par induction :

  (base) $S \cup \{T, \perp\}$

  (cons) Soit P et Q des mots de $(S \cup C \cup D \cup L)^*$, on dispose de 5 règles de construction

  $r1(P) = \neg P$

  $r2(P,Q) = (P \wedge Q)$

  $r3(P,Q) = (P \vee Q)$

  $r4(P,Q) = (P \rightarrow Q)$

  $r5(P,Q) = (P \leftrightarrow Q)$

Remarques

- Convention de ce cours :
  - Les majuscules dénotent des propositions (fbf):

    P,Q,R…
  - Les minuscules dénotent des symboles propositionnels

    p, q, r
- Attention :
  - Tout symbole propositionnel est une proposition

    p est à la fois une fbf de PROP({p,q}) et un symbole de {p,q}
  - Le contraire n'est pas vrai !

    $(p \wedge q)$ est une fbf de PROP({p,q}) mais n'est pas un symbole
  - Une formule réduite à un symbole propositionnel ou aux constantes logiques T ou $\perp$ est appelée une formule atomique ou atome

Différentes syntaxes

- Différentes fbf peuvent représenter les « mêmes conditions de vérité » (avoir la même sémantique)

  ➔ les connecteurs sont redondants

  Ex. $(P \leftrightarrow Q) \equiv ((P \rightarrow Q) \wedge (Q \rightarrow P))$

- Dans les démonstrations, on pourra se limiter aux seuls connecteurs $\neg, \wedge$ en traitant les autres comme des macros *(i.e. des raccourcis d'écriture)* :
  - $\perp$ pour $(P \wedge \neg P)$
  - T pour $\neg\perp$
  - $(P \vee Q)$ pour $\neg(\neg P \wedge \neg Q)$
  - $(P \rightarrow Q)$ pour $(\neg P \vee Q)$

Notions utiles

- L'ensemble des symboles propositionnels d'une fbf.

Si P est de la forme ¬Q

Ex. $\neg(p \wedge (q \rightarrow \neg r))$

$P = \neg\, Q$

$\mathcal{SP}$: PROP(S) → 2^S

(base) si $P \in S$, $\mathcal{SP}(P) = \{P\}$

si $P = \top$ ou $P = \bot$, $\mathcal{SP}(P) = \{\}$

(cons)

r1 : si $P = \neg Q$, $\mathcal{SP}(P) = \mathcal{SP}(Q)$

r2, r3, r4, r5 :

si $P=(Q \wedge R)$ ou $P=(Q \vee R)$ ou $P=(Q \rightarrow R)$ ou $P=(Q \leftrightarrow R)$,

$\mathcal{SP}(P) = \mathcal{SP}(Q) \cup \mathcal{SP}(R)$

Notions utiles (suite)

- Nombre de connecteurs d'une fbf :

  nbc: PROP(S) → N

  (base) nbc(P) = 0

  (cons)

  r1 : P=¬Q, nbc(P) =1+nbc(Q)

  autres règles : P=(Q c R) avec c connecteur binaire

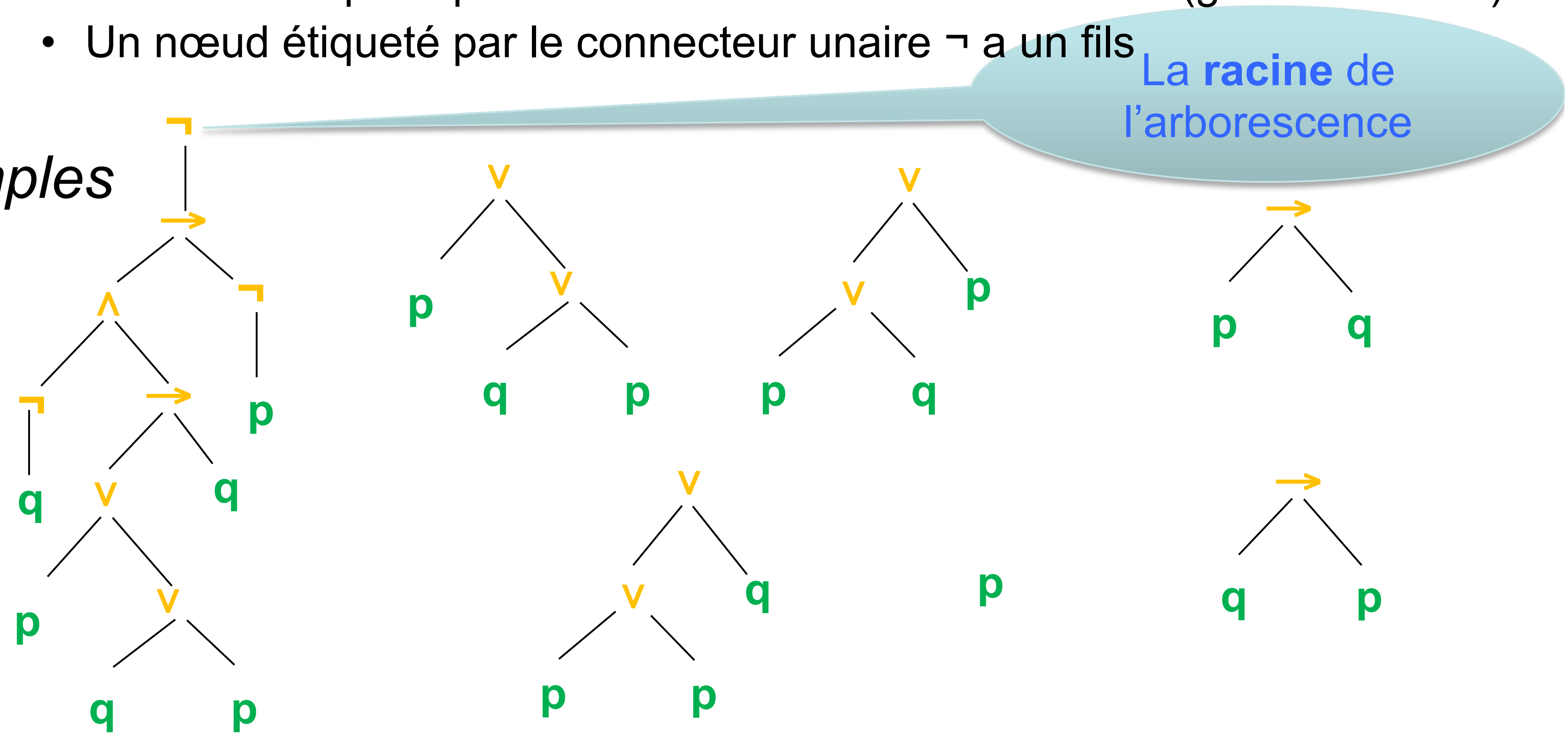
  nbc(P) = 1+nbc(Q)+nbc(R)

- *Exercice*
  - *Ens. des connecteurs*
  - *Ens. des sous-fbfs :*
    - *Sous-fbf : une partie d'une fbf elle-même une fbf (qui corresponde à une sous-arborescence)*

Notions utiles (suite)

- Soit ARBO(S) l'ensemble des arborescences dont
  - les **feuilles** sont étiquetées par des éléments de $S \cup \{T, \perp\}$, et
  - les **autres nœuds** par des connecteurs avec respect de l'arité
    - Un nœud étiqueté par un connecteur binaire a deux fils (gauche et droit)
    - Un nœud étiqueté par le connecteur unaire ¬ a un fils

Exemples



Notions utiles (fin)

- Propriété : A toute fbf correspond une unique arborescence (et vice versa) :
  - Soit fbf2arb (l'arbor. associée à une fbf) : PROP(S) → ARBO(S)

    (base) fbf2arb(P)= arbor. réduite à un sommet étiqueté par P

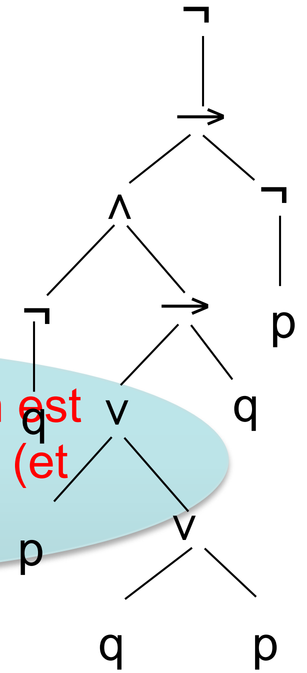
    (cons)
    r1: P=¬Q, fbf2arb(P)= arbor. de racine étiquetée par ¬ ayant comme unique fils la racine de fbf2arb(Q)

    autres r : P=(Q c R), fbf2arb(P)= arbor. de racine étiquetée par c ayant comme fils gauche la racine de fbf2arb(Q) et comme fils droit la racine de fbf2arb(R)

  - On montre que fbf2arb est une bijection

    - *Exercice : dessiner fbf2arb((A∧¬B)∧A))*
    - *Exercice : définir arb^{-1} la fonction qui a une arbor. associe une fbf*
    - *Exercice : définir profondeur d'une fbf*

Conventions



- Si les arbres montrent bien la structure d'une fbf, ils ne sont pas « pratique » pour l'écriture

Seule cette convention est admise dans ce cours (et acceptée à l'exam)

  – La notation **infixée** est la manière classique d'écrire les fbf mais elle nécessite l'utilisation de parenthèses

$$\neg((\neg q \wedge ((p \vee (q \vee p)) \rightarrow q)) \rightarrow \neg p)$$

  – On peut par des conventions de priorité de connecteurs éliminer des parenthèses

¬ est *prioritaire* sur ∧ et ∨ qui sont prioritaires sur → et ↔

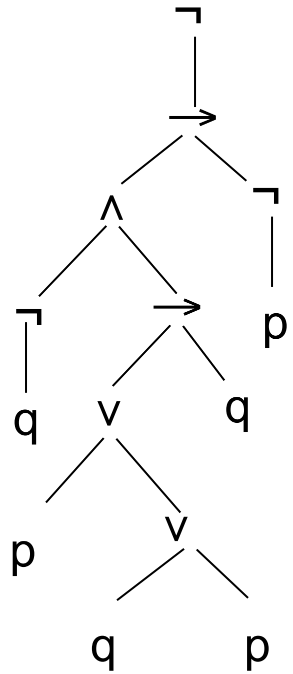
$((p \vee (p \rightarrow q)) \rightarrow \neg p)$ devient par convention $p \vee (p \rightarrow q) \rightarrow \neg p$

voire privilégier l'évaluation gauche-droite pour les connecteurs de même priorité

ici cela ne changerait rien, il faut conserver $p \vee (p \rightarrow q) \rightarrow \neg p$

Mais par exemple $(p \rightarrow ((p \vee q) \wedge p))$ deviendrait $p \rightarrow p \vee q \wedge p$

Conventions



- Si les arbres montrent bien la structure d'une fbf, ils ne sont pas « pratique » pour l'écriture
  - Les notations **préfixée** ou **post-fixée** ont l'avantage de ne pas nécessiter de parenthèses

    PRE (Rac, FG, FD) : ¬ → ∧ ¬ q → ∨ p ∨ q p q ¬ p

    POST (FG, FD, Rac) : q ¬ p q p ∨ ∨ q → ∧ p ¬ → ¬

  - En TP on utilisera une notation préfixée parenthèsée représentée par des listes pour faciliter l'utilisation des formules.

[¬ ; [→ ; [∧ ; [¬ ; q] ; [→ ; [∨ ; p ; [∨ ; q ; p]] ; q]] ; [¬ ; p]]]