Memo InfoT

P v !P => A ^ !A -> Universellement faux (vrai implique faux impossible)

(SF => AE) => (¬AE => ¬SF ) -> Universellement valide (contraposée)

0-résolution

- Faire !conclusion

- Remplacer => par v

- Vérifier si valide ou pas -> chercher une forme avec ce dont on dispose

Ex :

• MgO+H2→Mg+H2O

• C+O2 →C02• CO2 + H2O → H2CO3

On suppose que l’on dispose de MgO, H2, O2 et C

-> Mettre sous forme avec que MgO, H2, O2 et C

\_|\_ = contradiction

(!A ^ B) v C

-> !A v C et B v C

((A∨Φ1)∧(¬A∨Φ2))⊢(Φ1 ∨Φ2)

==> Chercher forme !A ^ A -> impossible

Si une seule proposition, faire sa négation

Prenexe:

- Enlever les variables doublons Ex: ∀x(p(x)=>∃x q(x)) --> ∀x(p(x)=>∃y q(y))

- Supprimer les => et mettre toutes les variables devant Ex: ∀x ∃y(!p(x) v q(y))

Skolem:

y est une fonction de x -> y = f(x)

Ex: !p(x) v q(f(x)))

/!\ PAs d'unicité des formes de Skolem (variables parfois faussement dépendantes)

Si aucun ∀ -> x0, etc. (constantes)

Unification:

Ex:

+ grand unificateur: [y | f (b); x | g(a, b); z | g(a, b)]+ grand unificateur: [y | f (g(v, w)); x | g(v, w); z | g(x, y)]

Opérations nombres

25 en base 2:

25/2 = 12 r 1

12/2 = 6 r 0

6/2 = 3 r 0

3/2 = 1 r 1

1/2 = 0 r 1

--> 25 = 11001

Conversion directe possible que de multiple de 2^n à 2^m

A5C = 10 + 5 + 12 = 1010 0101 1100

Octal = 3 bits -> A5C = 101 001 011 100

Valeur max: b^k -1

Nombre de valeurs ≠: b^k

Positif:

signe et grandeur, complément à un, complément à deux tous identiques

Négatif:

Signe et grandeur: 1er bit à 1

Complément à un: inversion de tous les bits

Complément à deux: complément à un + 1

0 positif = 0 complément à deux -> unicité d'écriture

/!\ 1er bit ne peux pas servir pour valeur positive -> 8 bits: -128 à 127

Overflow -> opérandes mêmes signes

• m bits pour la partie entière• n bits pour la partie fractionnaire

• La plus petite valeur que l’on peut coder : -2^m• La plus grande 2^m-2^-n

0 pas compté comme un réel positif

Pseudo mantisse:

2^n -1 = max

Exposant = max - décalage de la mantisse

0,0375

x2 = 0,075 -> 0

x2 = 0,15 -> 0

x2 = 0,3 -> 0

x2 = 0,6 -> 0

x2 = 1,2 -> 1

x2 = 0,4 -> 0

x2 = 0,8 -> 0

x2 = 1,6 -> 1

--boucle-- on veut 5 bits donc on finit

x2 = 1,2 -> 1

x2 = 0,4 -> 0

0,5

0,25

0,125

Exposant: pas tous des 1

1,11111 x 2^7 = (2-2^-5) x 2^7 = 252