FUNZIONI RICORSIVE PRIMITIVE:

= PIÙ PICCOLA CLASSE DI FUNZIONI CHE CONTIENE:
- FUNZIONI BASE
- OPERATORE "COMPOSIZIONE"
- OPERATORE "RICORSIONE PRIMITIVA"

chiusa rispetto agli operatori:

· FUNZIONI DI BASE:

- ZERO: $\lambda x_1 — \lambda x_n . Z = 0$
- SELEZIONE: $\lambda x_1 — \lambda x_n . U_i^n = x_i$
- SUCCESSORE: $\lambda x . S = x+1 = x'$

· COMPOSIZIONE:

siano

$\underbrace{\lambda x_1 — \lambda x_n}_{n \text{ variabili}} . g_1$
$\lambda x_1 — \lambda x_n . g_2$
$\vdots$
$\lambda x_1 — \lambda x_n . g_m$

m funzioni g

$\underbrace{\lambda x_1 ——— \lambda x_m}_{m \text{ variabili}} . h$

funzioni RICORSIVE PRIMITIVE
allora la funzione definita mediante l'OPERATORE di COMPOSIZIONE

$$h\left(g_1(x_1 — x_n), ———, g_m(x_1 — x_n)\right) = f(x_1 — x_n)$$

è RICORSIVA PRIMITIVA

· RICORSIONE:

siano

$\underbrace{\lambda x_1 — x_n}_{n \text{ variabili}} . g$
$\underbrace{\lambda y, \lambda x_1 — x_n, \lambda z}_{n+2 \text{ variabili}} . h$

funzioni ricorsive primitive
allora la funzione $\lambda x_1 — x_n, \lambda y . f$ ottenuta mediante l'OPERATORE di RICORSIONE

$$\begin{cases} f(0, x_1 — x_n) = g(x_1 — x_n) \\ f(\underbrace{n+1}_{\text{passo}}, \underbrace{x_1 — x_n}_{\text{input}}) = h(\underbrace{n}_{\text{passo precedente}}, \underbrace{x_1 — x_n}_{\text{input}}, \underbrace{f(n, x_1 — x_n)}_{\text{funzione f del passo precedente}}) \end{cases}$$

è RICORSIVA PRIMITIVA

- Somma

$$\begin{cases} +(a,0)=a & \Rightarrow U^2_1(a,0) \\ +(a,b')=(+(a,b))' & \Rightarrow S(+(a,b)) \end{cases}$$

- Prodotto

$$\begin{cases} \cdot(a,0)=0 & \Rightarrow U^2_2(a,0) \\ \cdot(a,b')=+(\cdot(a,b),a) \end{cases}$$

- esponenziale $(esp = a^b)$

$$\begin{cases} esp(a,0)=1=S(U^2_2(a,0)) \\ esp(a,b')=\cdot(esp(a,b),a) \end{cases}$$

- predecessore

$$\begin{cases} pred(0)=0=U^1_1(0) \\ pred(b')=b \end{cases}$$

- fattoriale

$$\begin{cases} fatt(0)=S(U^1_1(0)) \\ fatt(b')=\cdot(fatt(b),S(b)) \end{cases}$$

$\hookrightarrow b! = (b-1)!\cdot b$

- sottrazione chiusa $\left(a \dot{-} b = \begin{cases} 0 & \text{se } a \leq b \\ a-b & \text{se } a>b \end{cases}\right)$

$$\begin{cases} \dot{-}(a,0)=0=U^2_1(a,0) \\ \dot{-}(a,b')=pred(\dot{-}(a,b)) \end{cases}$$

- minimo

$min(a,b)=\dot{-}(b,\dot{-}(b,a)) \rightarrow b \dot{-} (b \dot{-} a)$

$b=3,\ a=2 \Rightarrow 3 \dot{-} (3 \dot{-} 2) = 3 \dot{-} 1 = 2$

$b=2,\ a=3 \Rightarrow 2 \dot{-} (2 \dot{-} 3) = 2 \dot{-} 0 = 2$

- massimo $\rightarrow a + (b \dot{-} a)$

$b=3,\ a=2 \Rightarrow 2+(3 \dot{-} 2) = 2+1 = 3$

$b=2,\ a=3 \Rightarrow 3+(2 \dot{-} 3) = 3+0 = 3$

$max(a,b)=+(a,\dot{-}(b,a))$

- segno $\left(\begin{cases} sg(0)=0 \\ sg(a')=1 \end{cases}\right)$

$sg(a)=1 \dot{-} \overline{sg}(a)$

- $\overline{\text{segno}}$ $\left(\begin{cases} \overline{sg}(0)=1 \\ \overline{sg}(a')=0 \end{cases}\right)$

$\overline{sg}(a)=1 \dot{-} a$

valore assoluto $|a-b|$

$b=2,\ a=3 \Rightarrow (3 \dot- 2) + (2 \dot- 3) = 1 + 0 = 1$

$b=3,\ a=2 \Rightarrow (2 \dot- 3) + (3 \dot- 2) = 0 + 1 = 1$

$$abs(-(a,b)) = +(\dot-(a,b), \dot-(b,a))$$

- resto (resto della divisione $b:a$)

$$\begin{cases} rm(a,0) = 0 = U_2^2(a,0) \\ rm(a,b') = S(rm(a,b)) \cdot sg(|a - S(rm(a,b))|) \end{cases}$$

- divisione (divisione $b':a$)

$$\begin{cases} qt(a,0) = 0 = U_2^2(a,0) \\ qt(a,b') = qt(a,b) + \overline{sg}(|a - S(rm(a,b))|) \end{cases}$$

- if-then-else

if $(A=0)$ then B else C $\Rightarrow \overline{sg}(A)\cdot B + sg(A)\cdot C$

$A=0 \Rightarrow 1\cdot B + 0 \cdot C \Rightarrow B$

$A\neq 0 \Rightarrow 0\cdot B + 1\cdot C \Rightarrow C$

$$+(\cdot(\overline{sg}(a), b), \cdot(sg(a), c))$$

$\Rightarrow$ sommatoria e produttoria $\Rightarrow$ vedi esercizio ricorsione.

SOMMATORIA (Σ)

se $\lambda x \lambda y \lambda z . f$ è RICORSIVA PRIMITIVA
allora

$$g(x,z) = \sum_{y=0}^{z} f(x,y,z)$$

è RICORSIVA PRIMITIVA.

<u>dim.</u>

$$\begin{cases} g^*(x,0,z) = f(x,0,z) \\ g^*(x,w+1,z) = \underbrace{g^*(x,w,z)} + f(x,w+1,z) \end{cases}$$

da cui:

$$g^*(x,w,z) = \sum_{y=0}^{w} f(x,y,z)$$

quindi:

$$g(x,z) = g^*(x,z,z)$$ è RICORSIVA PRIMITIVA

- PRODUTTORIA (Π)

se $\lambda xyz . f$ è RICORSIVA PRIMITIVA
allora

$$g(x,z) = \prod_{y=0}^{z} f(x,y,z)$$

è RICORSIVA PRIMITIVA

<u>dim.</u>

$$\begin{cases} g^*(x,0,z) = f(x,0,z) \\ g^*(x,w+1,z) = \underbrace{g^*(x,w,z)} \cdot f(x,w+1,z) \end{cases}$$

da cui

$$g^*(x,w,z) = \prod_{y=0}^{w} f(x,y,z)$$

quindi

$$g(x,z) = g^*(x,z,z)$$ è RICORSIVA PRIMITIVA

PREDICATI e CONNETTIVI LOGICI

- predicati

n-ari = relazione n-aria tra numeri naturali

$P: \mathbb{N}^n \to \{T, F\}$

un PREDICATO è RICORSIVO PRIMITIVO $\iff$ $\exists$ una funzione f RICORSIVA PRIMITIVA che lo calcola. (= FUNZIONE CARATTERISTICA)

$\forall \bar{z}^p$

$P(\bar{z}^p) = \text{VERO} \iff f(\bar{z}^p) = 0$

$P(\bar{z}^p) = \text{FALSO} \iff f(\bar{z}^p) = 1$

- not

$\neg P \to \overline{sg}(f_P)$

P	$\neg P$	f_P	$\overline{sg}(f_P)$
V	F	0	1
F	V	1	0

- and

$P \wedge Q \to sg(f_P + f_Q)$

P	Q	f_P	f_Q	$f_P + f_Q$	$sg(\cdot)$	$P \wedge Q$
V	V	0	0	0	0	V
V	F	0	1	1	1	F
F	V	1	0	1	1	F
F	F	1	1	2	1	F

- or

$P \vee Q \to sg(f_P \cdot f_Q)$

P	Q	f_P	f_Q	$f_P \cdot f_Q$	$sg(\cdot)$	$P \vee Q$
V	V	0	0	0	0	V
V	F	0	1	0	0	V
F	V	1	0	0	0	V
F	F	1	1	1	1	F

- $\Rightarrow$

$P \Rightarrow Q = \neg P \vee Q$ (si dimostra per composizione.)

- $\bigvee$ (= GRANDE "or")

$\bigvee_{i=1}^{n} P_i = \prod_{i=1}^{n} f_{P_i}$ $\quad (= \exists\ 0 \le x \le n\ P_x)$

- $\bigwedge$ (= GRANDE "and")

$\bigwedge_{i=1}^{n} P_i = \sum_{i=1}^{n} f_{P_i}$ $\quad (= \forall\ 0 \le x \le n\ P_x)$

$f\left(\begin{array}{l}\to \text{VERO se } x \neq y \ (=0) \\ \to \text{FALSO se } x = y \ (=1)\end{array}\right)$

$x \neq y \to \overline{sg}(|x \dot- y|)$

$x=3 \ y=2 \quad \overline{sg}|x-y| = \overline{sg}((x \dot- y)+(y \dot- x)) = \overline{sg}(1+0) = 0$

$x=2 \ y=3 \quad \overline{sg}|x-y| = \overline{sg}((x \dot- y)+(y \dot- x)) = \overline{sg}(0+1) = 0$

$x=2 \ y=2 \quad \overline{sg}|x-y| = \overline{sg}((x \dot- y)+(y \dot- x)) = \overline{sg}(0) = 1$

- $=$

$x=y \quad sg(|x-y|)$

- $\leq$

$x \leq y \to \bigvee_{z=0}^{y} x+z=y$

$\to \exists z$ con $0 \leq z \leq y$

- / (= è divisibile)

$x/y \to \bigvee_{z=0}^{y} x \cdot z = y$

(= y è divisibile per x)

$\to \exists z$ con $0 \leq z \leq y$

- **numero primo** (= definizione ricorsiva primitiva)

$\underbrace{x \neq 0 \wedge x \neq 1}_{\text{se è un numero primo è minimo 2}} \wedge \underbrace{\bigwedge_{z=1}^{x}}_{\forall z,\ 1 \leq z \leq x} (\text{se } \underbrace{z|x}_{\text{se z divide x}} \text{ allora } \underbrace{z=1 \vee z=x}_{z \text{ o è } =1 \text{ opp } =x})$

$\Rightarrow$ (= numeri primi sono divisibili o per se stessi o per uno.)

- **pari**

x è pari $\to 2|x$

- **dispari**

x è dispari $\to \neg$ pari

FUNZIONI RICORSIVE PARZIALI:

= più piccola classe di funzioni che contiene:

- funzioni di base
- chiusa rispetto agli operatori:
  - composizione
  - ricorsione primitiva
  - minimalizzazione

• MINIMALIZZAZIONE LIMITATA:

data una funzione $\lambda\bar{x}^p, \lambda\tau . g$ con $0 \leq \tau \leq 4$

si definisce la funzione $\lambda\bar{x}^p . f$ mediante l'operatore di minimalizzazione μ

$$\lambda\bar{x}^p . f = \mu\tau\left(g(\bar{x}^p, \tau) = 0\right)$$ (= minimo τ che azzera la funzione g)

se $g(\bar{x}^p, \tau) \neq 0$, $\forall\tau$, allora f è indefinita

⇓

l'operatore μ è ricorsivo primitivo

dim.

consideriamo: → si termina quando $sg(\cdot) = 0 \iff g(\bar{x}, \tau) = 0$ ~~[illegible]~~

$\boxed{sg(g(\bar{x}, 0))} +$

→ per proseguire uso il fatto che lo trovato lo 0 moltiplico

$sg(g(\bar{x}, 0)) \cdot sg(g(\bar{x}, 1)) +$

$\vdots$

continuo a calcolare i prodotti, [illegible] righe T^p tante = a [illegible] una che $\bar{x} = 0$ che è proprio il minimo. per sapere il valore di T sommo tutti gli 1.

$$sg(g(\bar{x}, 0)) \cdot sg(g(\bar{x}, 1)) \cdot \dots \cdot sg(g(\bar{x}, 4)) =$$

$$= \sum_{i=0}^{4} \prod_{j=0}^{i} sg(g(\bar{x}, j))$$ che è ricorsiva primitiva

⇓

poiché esiste una funzione ricorsiva che lo calcola, l'operatore di minimalizzazione è ricorsivo primitivo.

$/$ (= DIVISORE)

$$x \mid y = \mathop{M}_{z=0}^{x} \; yz' = x$$

(x divide y)

$y : x = z'$

- numero primo

$$\begin{cases} p(0) = 2 \\ p(n') = \mathop{M}\limits_{z=0}^{p(n)!+1} \left(p(z) \wedge p(n) < z \right) \end{cases}$$

- massimo esponente

$$exp(n,x) = \mathop{M}_{z=0}^{x} \neg\, p(n)^{z+1} \mid x$$

(= massimo esponente possibile del numero primo p(n) nella scomposizione in fattori primi del numero x)

// REGOLE DEI TABLEAU:

- $$\begin{array}{c} \neg\neg\varphi \\ | \\ \varphi \end{array}$$

- $$\begin{array}{c} \varphi \wedge \psi \\ | \\ \varphi \\ \psi \end{array}$$] sono veri ;] se entrambi sono veri

  $$\begin{array}{ccc} & \neg(\varphi \wedge \psi) & \\ \swarrow & & \searrow \\ \neg\varphi & & \neg\psi \end{array}$$] sono veri ;] se almeno uno è vero

- $$\begin{array}{ccc} & \varphi \vee \psi & \\ \swarrow & & \searrow \\ \varphi & & \psi \end{array}$$

  $$\begin{array}{c} \neg(\varphi \vee \psi) \\ | \\ \neg\varphi \\ \neg\psi \end{array}$$

- $$\begin{array}{ccc} & \varphi \Rightarrow \psi & \\ \swarrow & & \searrow \\ \neg\varphi & & \psi \end{array}$$ $(= \neg\varphi \vee \psi)$

  $$\begin{array}{c} \neg(\varphi \Rightarrow \psi) \\ | \\ \varphi \\ \neg\psi \end{array}$$ $(= \neg(\neg\varphi \vee \psi) = \varphi \wedge \neg\psi$

- $$\begin{array}{ccc} & \varphi \Leftrightarrow \psi & \\ \swarrow & & \searrow \\ \neg\varphi & & \psi \\ \neg\psi & & \varphi \end{array}$$ $\left(= (\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi)\right)$

  $$\begin{array}{ccc} & \neg(\varphi \Leftrightarrow \psi) & \\ \swarrow & & \searrow \\ \varphi & & \psi \\ \neg\psi & & \neg\varphi \end{array}$$ $\left(= \neg((\neg\varphi \vee \psi) \wedge (\neg\psi \vee \varphi))\right.$
  $= \neg(\neg\varphi \vee \psi) \vee \neg(\neg\psi \vee \varphi) =$
  $= (\varphi \wedge \neg\psi) \vee (\psi \wedge \neg\varphi)$

TABLEAU CON QUANTIFICATORI:

$$\begin{array}{c} \forall x . \varphi \\ | \\ \varphi(t) \end{array}$$
└ simbolo usato precedentem.

$$\begin{array}{c} \neg\forall x . \varphi \\ | \\ \neg\varphi(a) \end{array}$$
└ simbolo non usato precedentem.

$$\begin{array}{c} \exists x . \varphi \\ | \\ \varphi(a) \end{array}$$

$$\begin{array}{c} \neg\exists x . \varphi \\ | \\ \neg\varphi(t) \end{array}$$

ESERCIZI coi TABLEAU:

$[[P\to R]\wedge[Q\to R]]\to[[P\vee Q]\to R]$ — si vuole provare che questa tautologia è falsa.

F

V F

V V V F

P=F R=F Q=F R=F (F) → R=F

→ P VERO or Q VERO => contraddizione => è una tautologia

$[P\to[Q\to R]]\to[[P\to Q]\to[P\to R]]$

F

V F

V F

P VERO Q VERO R FALSO =FALSO   P VERO Q VERO   P=VERO R=FALSO

=FALSO → contraddizione => è una tautologia

⇓

dimostriamo gli stessi con i TABLEAU

→ si vuole tutto per provare che non è una tautologia

V $\neg[[P\to[Q\to R]]\to[[P\to Q]\to[P\to R]]]$ — $\underbrace{[P\to[Q\to R]]}_{\varphi}$, $\underbrace{[[P\to Q]\to[P\to R]]}_{\psi}$

$(\neg(\varphi\to\psi)=\neg(\neg\varphi\vee\psi)=\varphi\wedge\neg\psi$

$=\varphi\to\psi=\neg\varphi\vee\psi$ (si dimostra)

$=\varphi\wedge\neg\psi$

(regole dritte ⇓ risolvo prima quelle che si svolgono dritte)

V $[P\to[Q\to R]]$

V $\neg[[P\to Q]\to[P\to R]]$

V $P\to Q$

V $\neg[P\to R]$

V P

V $\neg R$ → ho esteso prima tutti quelli che si svolgono dritto. Ad ritroso e considero quelli che si diramano.

- Q
  - $Q\to R$
    - R
    - $\neg Q$
  - $\neg P$
- $\neg P$ → si chiude il ramo perché è un cammino che contiene P e $\neg P$

⇓

TUTTI I RAMI SONO CHIUSI

⇓

Γ È INCONSISTENTE, cioè $\forall$ STRUTTURA in Γ, $\exists\ \varphi\in\Gamma$ t.c. φ è FALSA

($\Gamma\models$) (SEMANTICAMENTE)

→ = insieme di formule

$\Gamma\not\models$

(=> se esiste almeno una struttura con tutte le formule vere Γ è semanticamente consistente)

$\checkmark\ \neg\Big[\big[[P\to R]\wedge[Q\to R]\big]\to\big[[P\vee Q]\to R\big]\Big]$

$\neg\varphi\to\psi$ / φ / $\neg\psi$

$\checkmark\ \big[[P\to R]\wedge[Q\to R]\big]$

$\checkmark\ \neg\big[[P\vee Q]\to R\big]$

→ scendono tutti e due diritti. li estendo.

$\varphi\wedge\psi$ / φ / ψ

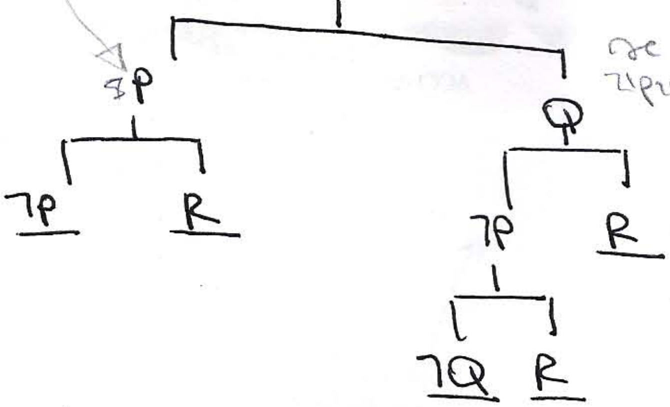
$\checkmark\ P\vee Q$

$\neg R$

$P\to R$

$Q\to R$

$\varphi\vee\psi$ / φ ψ

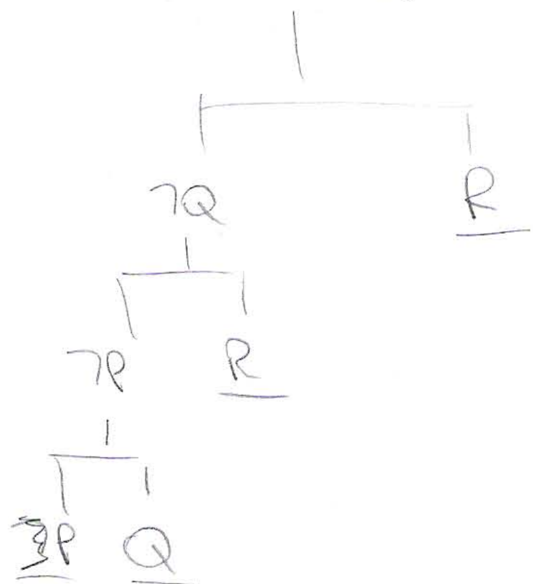


se
¬P va tolto il primo

ripeto per ogni ramo.
perché il valore di verità
dipende da entrambi

$\Gamma\cup\{\neg\varphi\}$

è inconsistente sintatticamente $\Gamma\vdash\varphi$



altrimenti posso fare come prima
dal ramo senza nulla.

IDEA TABLEAU:

se A è una tautologia, allora ¬A è inconsistente.
Partiamo da ¬A e semplifichiamo l'albero. Se troviamo ad un formula e una stessa negata
contraddizione chiudiamo il ramo.
Se per tutti gli stadi succede questo, tutti i rami chiusi, non è possibile falsificare la
formula, quindi A è una tautologia.

COMPLETENESS THEOREM:

se $\Gamma \models \varphi$ allora $\Gamma \vdash \varphi$
(= consistente semanticamente) (= consistente sintatticamente)

si assume $\Gamma \models$ e si prova $\Gamma \vdash$
(= inconsistente semanticamente) (= inconsistente sintatticamente)

si suppone di avere un tableau generato da Γ in cui tutte le formule composte sono già state etichettate di lato (= utilizzate) e quindi non ci sono più regole da utilizzare.

se il tableau è chiuso $\Rightarrow$ bene!

altrimenti si prende un ramo non chiuso e se appare un ~~formula~~ atomo non negato si dà valore TRUE altrimenti FALSE.

$\Downarrow$

Si ottiene quindi che lungo questo ramo tutte le formule in questa struttura sono vere.

ma questo è impossibile perché si è supposto $\Gamma \models$

quindi il tableau è chiuso e non ci sono rami aperti:

$\Gamma \models \quad \Gamma \vdash$

supponiamo ora che alcune formule lungo il ramo sono false e consideriamo la più corta.

la formula non può essere un atomo perché li abbiamo messi a True.

non può essere un atomo negato perché visto che gli atomi sono True, avrei in uno stesso ramo atomo e $\neg$(atomo) e tuttavia il ramo non sarebbe chiuso.

quindi la formula è composta (o al massimo è una doppia negazione) prendiamo l'ultima formula.

se la testa è falsa avremo in almeno uno dei rami una formula falsa. questa formula non può essere un atomo (perché sarebbe True), quindi è composta: ho trovato un'altra formula applicabile dopo quella che ritenevo essere l'ultima. $\Rightarrow$ assurdo

$\Downarrow$

nel tableau tutte le formule lungo un ramo ~~chiuso~~ aperto devono essere vere nella struttura.

$\Downarrow$

quindi $\Gamma \models \varphi$ allora $\Gamma \vdash \varphi$

SOUNDNESS THEOREM:

se $\Gamma \vdash \varphi$ allora $\Gamma \vDash \varphi$

(consistente sintatticamente) (consistente semanticamente)

è sufficiente provare che:

$\Gamma \vdash$ allora $\Gamma \vDash$

(inconsistente sintatticamente) (inconsistente semanticamente)

per assurdo supponiamo che:

$\Gamma \vdash$ e $\Gamma \vDash \varphi$ = cioè tutte le formule in Γ sono vere in una certa situazione.

(= consistente semanticamente)

$\Downarrow$

nel generare il tableau voglio che tutte le formule della situazione siano vere, ma anche tutte quelle che si incontrano sviluppando il ramo.

ci sono due modi per aggiungere formule nei tableau:

1) estensione senza diramazioni.
se l'antecedente è vero, si aggiungono tutte le formule vere al ramo

2) estensione con diramazione:
se l'antecedente è vero, uno dei rami che si creano ha solo formule vere.

$\Downarrow$

su tutti i rami ho solo formule vere, quindi nessun ramo si chiude (perché non può contenere formule contraddittorie)

$\Downarrow$

quindi se Γ è consistente semanticamente è anche consistente sintatticamente

$\Downarrow$

ma è un assurdo perché per ipotesi Γ è inconsistente sintatticamente, quindi Γ è inconsistente anche semanticamente.