PCAD
Programmazione Concorrente
Algoritmi Distribuiti

Arnaud Sangnier
arnaud.sangnier@unige.it

Verifica di sistemi 3

Logica Temporale Lineare (LTL)

- La logica LTL permette di descrivere delle proprietà sulle tracce di un sistema
- Una formula di LTL permette quindi di descrivere un sotto-insieme di $(2^{PA})^{\omega}$
- In particolare:
  - Una formula di LTL permette di parlare di uno stato particolare. Per esempio, se abbiamo PA={a,b,c} allora la formula ((a and not b) or c) descrive tutti gli stati in cui l'etichetta c è presente oppure l'etichetta a è presente ma non l'etichetta b
  - Esempio di stati verificando ((a and not b) or c)
  - Una formula LTL permette anche di spostarsi lungo ad una traccia

Sintassi di LTL

- In LTL abbiamo due operatore temporale:

  1) **X φ** vuol dire '**Next** φ' per dire che si guardiamo lo stato seguente nella traccia allora soddisfa

  2) **φ U Ψ** vuole dire 'φ **Until** Ψ', la formula φ rimane vera fino che raggiungiamo uno stato dove Ψ è vera (e un tale stato è effetivamente raggiunto)

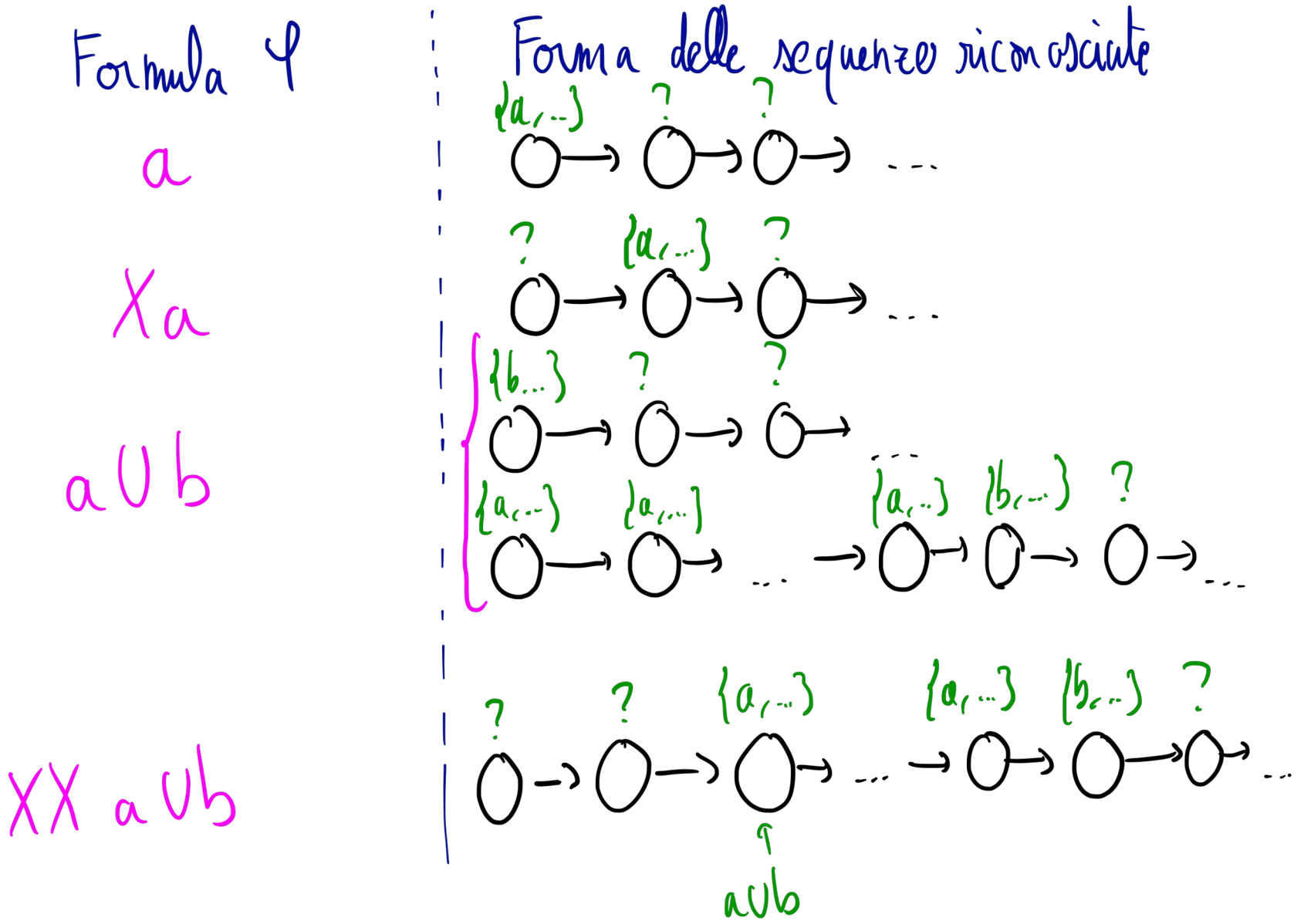
- Sintassi di LTL:

$$\varphi ::= true \mid a \mid \varphi \text{ and } \varphi \mid \text{not } \varphi \mid X\,\varphi \mid \varphi\ U\ \varphi$$

dove $a \in PA$

- Una formula di LTL rappresenta delle sequenze infinite in $(2^{PA})^{\omega}$ e dando una tale sequenza si sposta 'sopra'

Esempio


Formula φ
Forma delle sequenze riconosciute
a
Xa
a U b
X X a U b
{a,...}
?
{b,...}
a U b

Semantica di LTL

- Consideriamo una sequenza infinita $\sigma=\sigma_0\sigma_1\sigma_2...$ in $(2^{PA})^{\omega}$.
- Per tutti i, scriviamo $\sigma[i..]$ per la sequenza infinita $\sigma_i\sigma_{i+1}\sigma_{i+2}$ ...
- Sia φ una formula di LTL, definiamo $\sigma \models \varphi$ in un modo induttivo:
  - $\sigma \models$ true **è sempre vero**
  - $\sigma \models a$ **sse** $a \in \sigma_0$
  - $\sigma \models \varphi$ and Ψ **sse** $\sigma \models \varphi$ e $\sigma \models \Psi$
  - $\sigma \models$ not φ **sse** $\sigma \not\models \varphi$
  - $\sigma \models X\varphi$ **sse** $\sigma[1..] \models \varphi$
  - $\sigma \models \varphi \, U \, \Psi$ sse esiste $j \geq 0$ tale che $\sigma[j..] \models \Psi$ e per tutti i tale che $0 \leq i < j$, abbiamo $\sigma[i..] \models \varphi$
- Scriviamo $Seq(\varphi)=\{\sigma \text{ in } (2^{PA})^{\omega} \mid \sigma \models \varphi \}$
- Quindi $Seq(\varphi) \subseteq (2^{PA})^{\omega}$ quindi $Seq(\varphi)$ è una proprietà temporale lineare

Scorciatoie

- Scorciatoie classiche:
  - **φ or Ψ** ::= not (not φ and not Ψ)
  - **φ ⇒ Ψ** ::= (not φ) or Ψ
- Scorciatoie temporale:
  - **F φ** ::= true U φ (un giorno abbiamo φ)
  - **G φ** ::= not(F (not φ)) (abbiamo sempre φ)
- Formule classiche:
  - **GF φ**: la formula φ è vera infinitamente spesso
  - **FG φ**: un giorno φ diventa sempre vera

Esempio

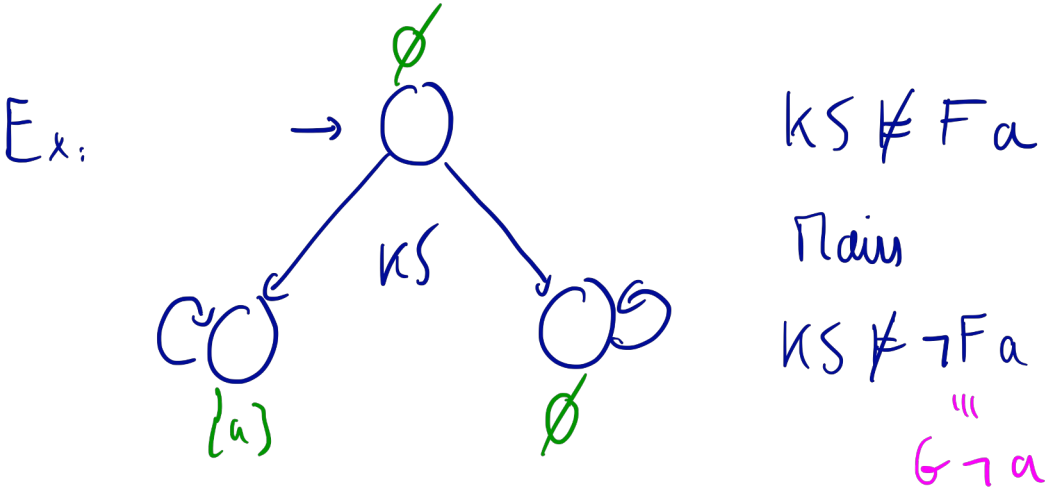
- Sistema con due processi che vogliono entrare in sezione critica e PA={crit1,crit2}
  - I due processi non sono mai allo stesso tempo in sezione critica:
    - G((not crit1) or (not crit2))
  - Ogni processo accede infinitamente spesso alla sua sezione critica:
    - GF(crit1) and GF(crit2)
    - **Attenzione:** è diverso da GF(crit1 and crit2)
- Sistema con un semaforo stradale e PA={V,G,R}:
  - Quando è verde, il semaforo non può diventare rosso direttamente nel passo successivo:
    - G(V ⇒ (not (X V)))
  - Quando è verde, il semaforo diventa giallo dopo un po' (e prima rimane verde), poi diventa rosso (e nel fra tempo è rimasto giallo)
    - G(V ⇒ (V U(G and X(G U R))))

Semantica delle scorciatoie

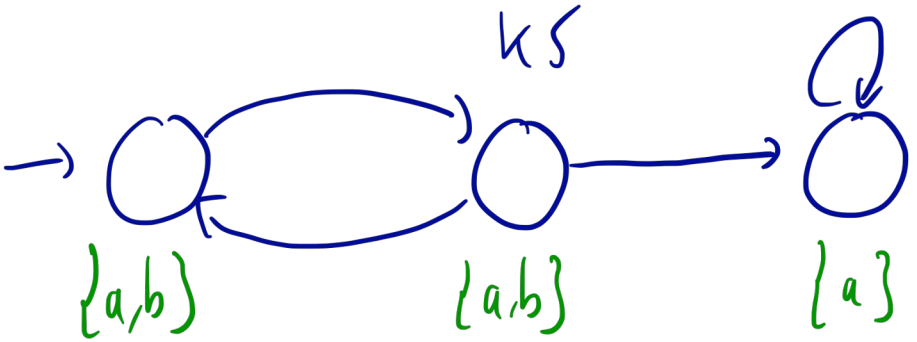
- $\sigma \models F\ \varphi$ **sse** esiste $j \geq 0$ tale che $\sigma[j..] \models \varphi$
- $\sigma \models G\ \varphi$ **sse** per tutti $j \geq 0$ $\sigma[j..] \models \varphi$
- $\sigma \models GF\ \varphi$ **sse** per tutti $j \geq 0$, esiste $k \geq j$ tale che $\sigma[k..] \models \varphi$
- $\sigma \models FG\ \varphi$ **sse** esiste $j \geq 0$ tale che per tutti $k \geq j$ $\sigma[k..] \models \varphi$

Model-checking di LTL

- Sia KS=(S,→, s_{in}, PA, L) una struttura di Kripke e una formula di LTL φ
- Si dice che KS soddisfa φ, scritto KS ⊨ φ sse Trace(KS) ⊆ Seq(φ)
- I.e. KS ⊨ φ sse **per tutte le tracce** σ di KS abbiamo σ ⊨ φ
- Come LTL è chiuso per negazione abbiamo KS ⊨ φ sse Trace(KS) ∩ Seq(not φ)= ∅,
- Attenzione: non abbiamo che se KS ⊭ φ allora KS ⊨ not φ



Esempio



- $KS \models X(a \text{ and } b)$
- $KS \models G\, a$
- $KS \models G((\text{not } b) \Rightarrow G(a \text{ and } (\text{not } b)))$

  $KS \not\models b \,U\, (a \text{ and } (\text{not } b))$

Model-checking in pratica

- Sia KS=(S,→, s_{in}, PA, L) una struttura di Kripke e una formula di LTL φ
- Sappiamo che se φ è una formula di LTL allora Seq(φ) è una proprietà temporale lineare regolare (la prova non è ovvia)
- Esiste quindi un automa di Büchi A_φ tale chee $L_\omega(A_\varphi) = Seq(\varphi)$
- Lo stesso vale per not φ i.e. esiste un automa di Büchi $A_{not\ \varphi}$ tale che $L_\omega(A_{not\ \varphi}) = (2^{PA})^\omega \setminus Seq(\varphi) = Seq(not\ \varphi)$
- Per verificare se KS⊨ φ, basta verificare se $Trace(KS) \cap L_\omega(A_{not\ \varphi}) = \varnothing$ usando il prodotto KS ⊗ $A_{not\ \varphi}$
  - Se $Trace(KS) \cap L_\omega(A_{not\ \varphi}) = \varnothing$ allora KS⊨ φ
  - Se $Trace(KS) \cap L_\omega(A_{not\ \varphi}) \neq \varnothing$ allora KS⊭ φ