

Tecniche di Monitoring di Servizi per la Soddisfazione Parziale dei Requisiti

MIRKO ZICHICHI

RELATORI: ING. MASSIMO COSSENTINO
 ING. LUCA SABATUCCI
 DOTT. GIADA DE SIMONE

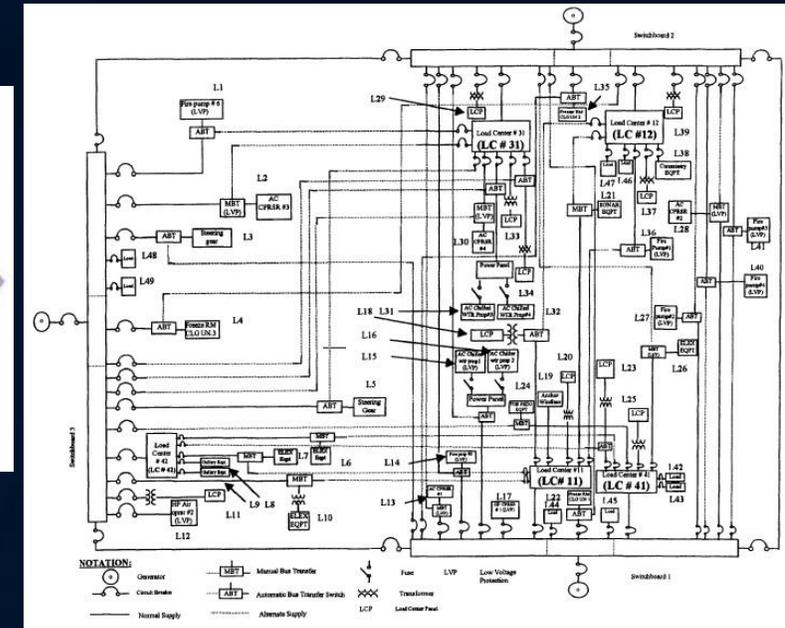
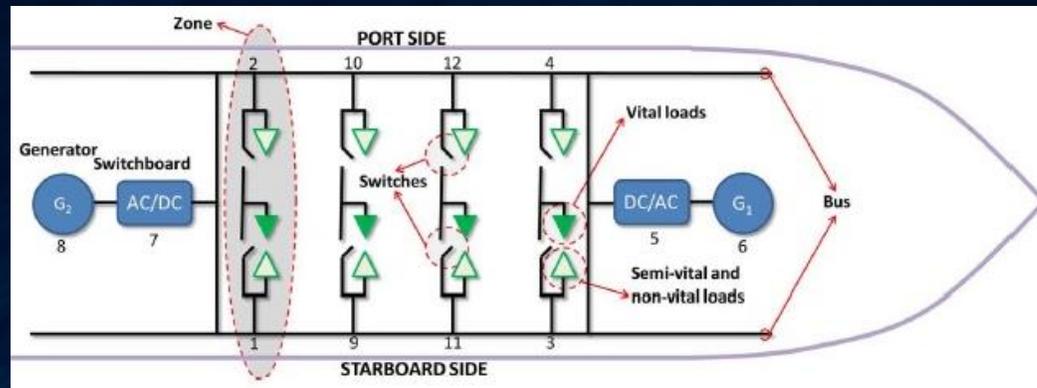
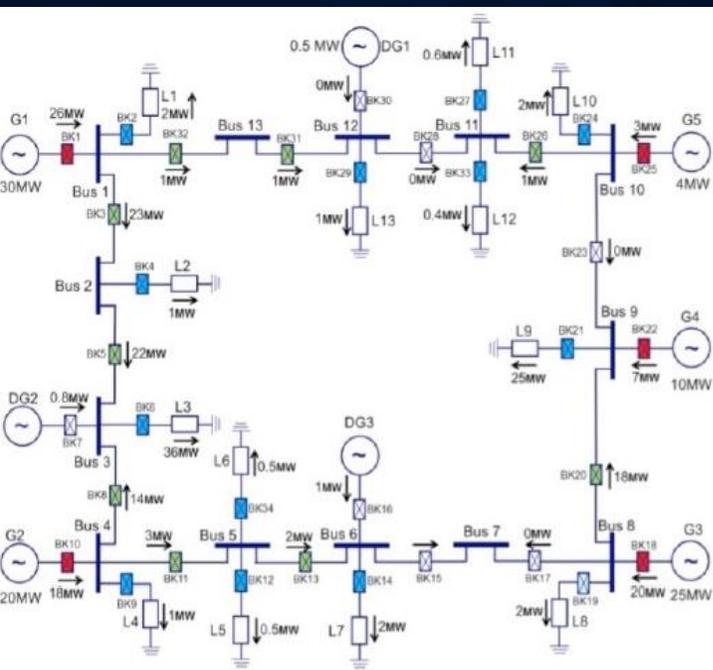
Sommario

- PROBLEMA
 - Caso di Studio
- SOLUZIONE PROPOSTA
 - Logica Temporale Lineare (LTL)
 - Reti di Petri
 - Monitor
- VALIDAZIONE
 - Esempio

Problema

Riconfigurazione di un Sistema di Alimentazione Elettrica di una Nave

- In seguito ad un guasto bisogna trovare una configurazione alternativa che renda possibile mantenere l'alimentazione elettrica, considerando fra i vari aspetti anche quello temporale.



Problema

Obbiettivi

- Se il Generatore Principale si guasta bisogna accendere il Generatore Ausiliario
- Il Carico 1 deve essere sempre alimentato
- Il Carico 2 deve essere alimentato finché è acceso il Motore 2
- Se si verifica un incendio bisogna staccare l'alimentazione al Carico 2 e accendere il sistema antincendio

Soluzione

- MUSA è un middleware basato su agenti per lo sviluppo di Sistemi Self-Adaptive guidati dagli utenti
 - Un Sistema Self-Adaptive è un insieme di più parti, indipendenti e non, che formano un'unica entità capace di rispondere a cambiamenti in maniera autonoma, modificando il suo comportamento o la sua struttura per raggiungere l'obiettivo fissato.
 - MUSA utilizza un approccio Goal-Oriented, le specifiche dell'utente sono descritte come un insieme di Goal.
- **Monitor per il soddisfacimento dei Goal**

Costruito tramite un insieme di Reti di Petri dipendenti tra di loro e supporta la Logica Temporale Lineare per la specifica dei Goals.

Logica Temporale Lineare (LTL)

LTL è una Logica Temporale che estende la Logica Proporzionale e permette di modellare il tempo tramite una successione infinita di Stati.

Connettivi Logici

$\neg \varphi$ $\varphi \vee \theta$ $\varphi \wedge \theta$ $\varphi \rightarrow \theta$ $\varphi \leftrightarrow \theta$

Operatori Temporal

X φ

NEXT

F φ

FINALLY

G φ

GLOBALLY

φ **U** θ

UNTIL

φ **R** θ

RELEASE

Es.

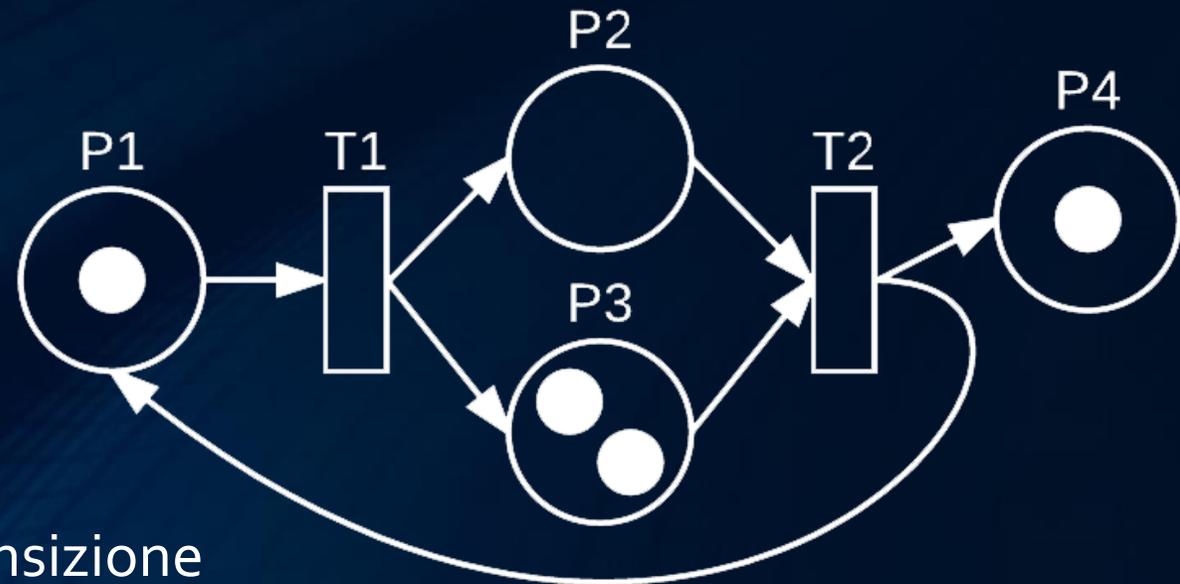


Rete di Petri

Una Rete di Petri è un linguaggio grafico e matematico, rappresentato da un grafo diretto, che permette la modellazione di un sistema distribuito.

Una Rete di Petri è una tupla $PN = \{P, T, A\}$ dove:

- P è l'insieme dei **Posti**
- T è l'insieme delle **Transizioni**
- $A \subseteq (P \times T) \cup (T \times P)$
è l'insieme degli **Archi**
- Ogni Posto può contenere uno o più **Token**
- Un Token passa da un Posto ad un altro attraverso lo **Scatto** di una Transizione



Monitor



Monitor



Monitor



Monitor



Monitor



Formula LTL (1/2)



$\theta : F (c U (a \wedge b))$

Formula LTL (1/2)



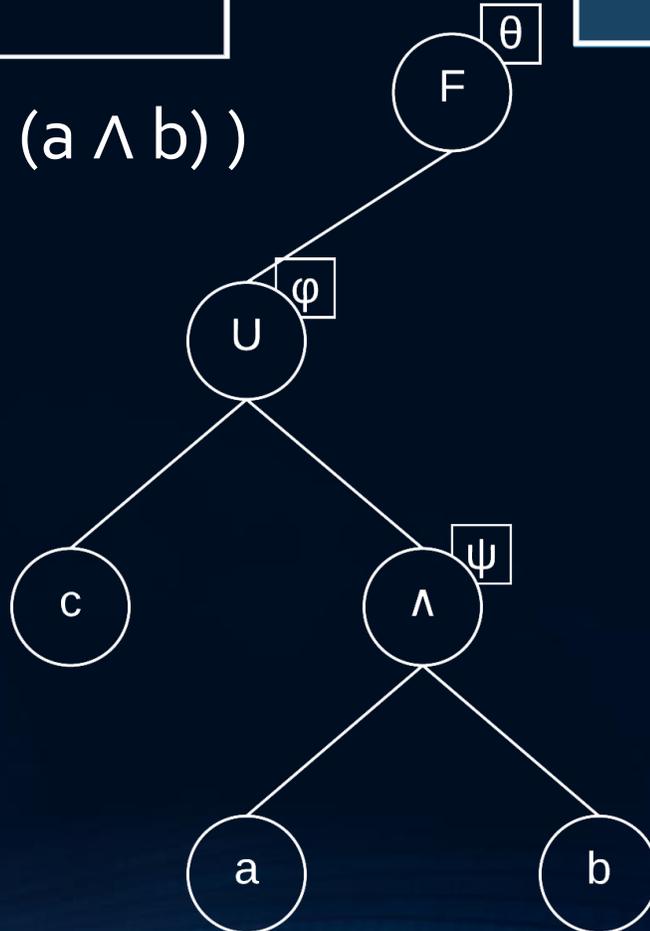
$\theta : F (c U (a \wedge b))$



Formula LTL (2/2)



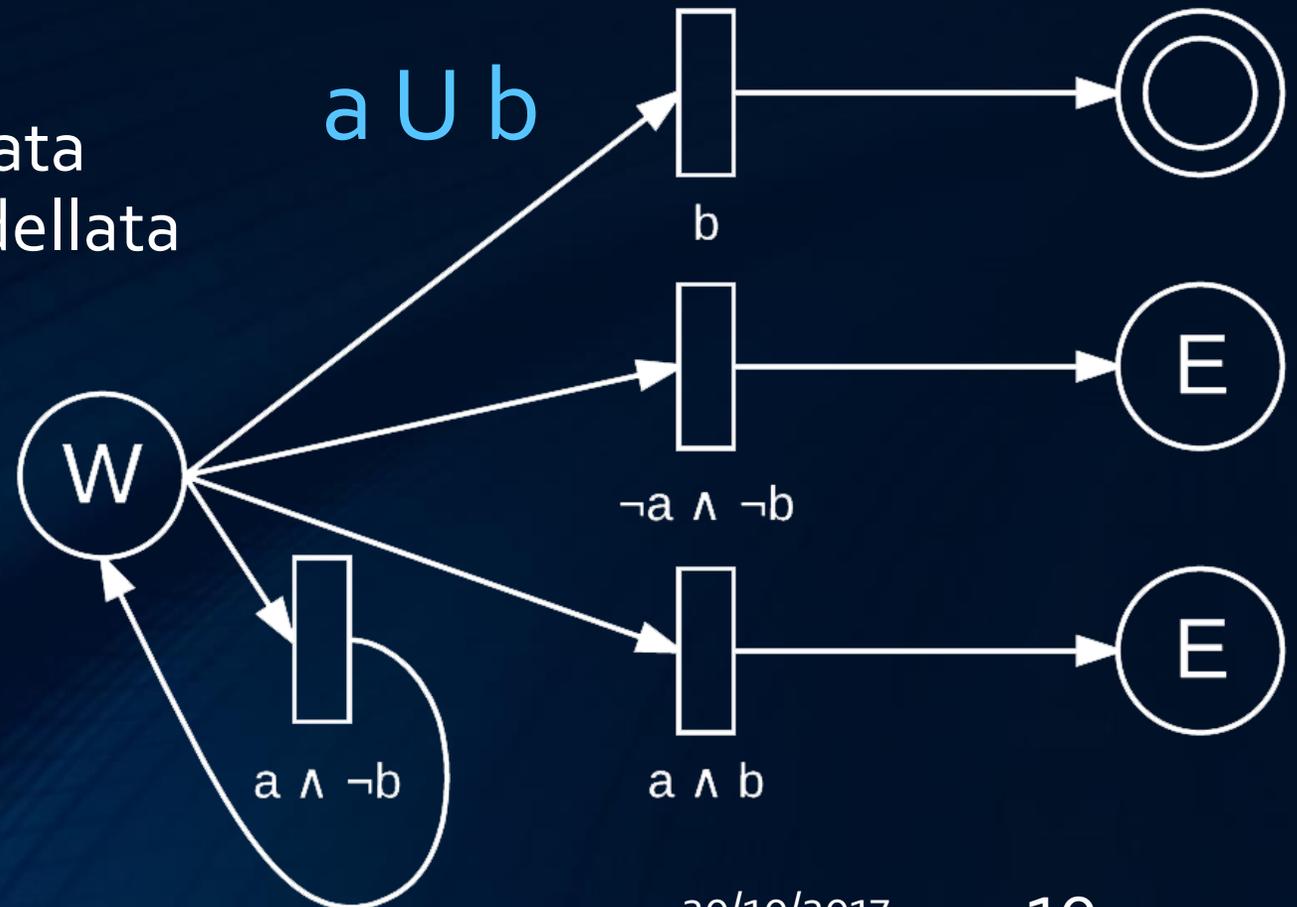
$\theta : F(c U (a \wedge b))$



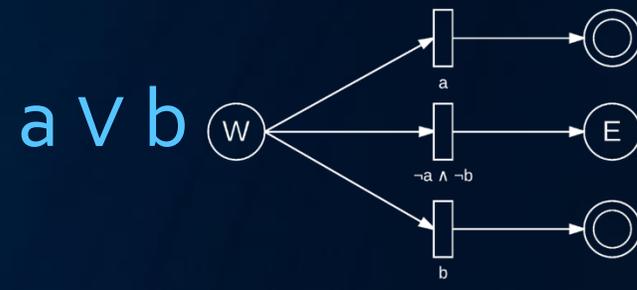
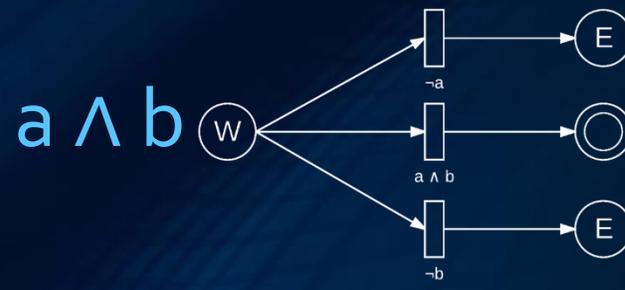
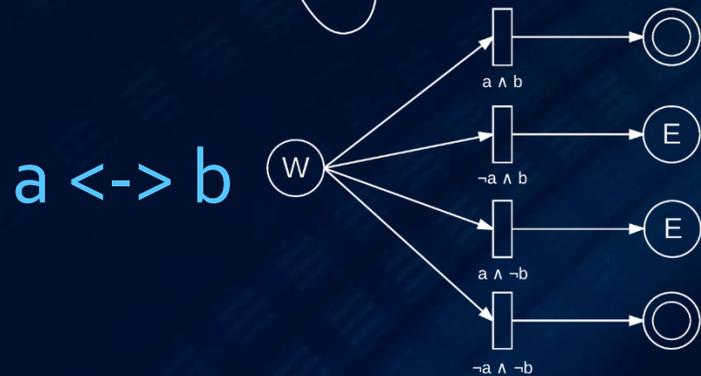
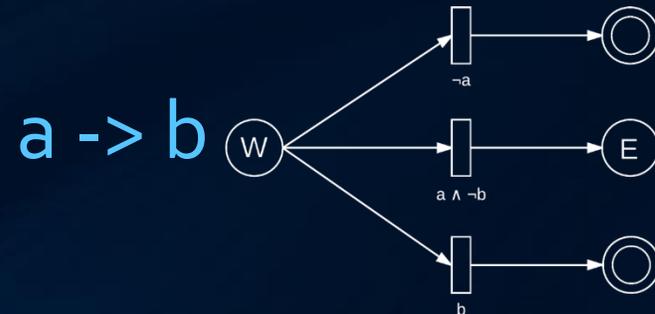
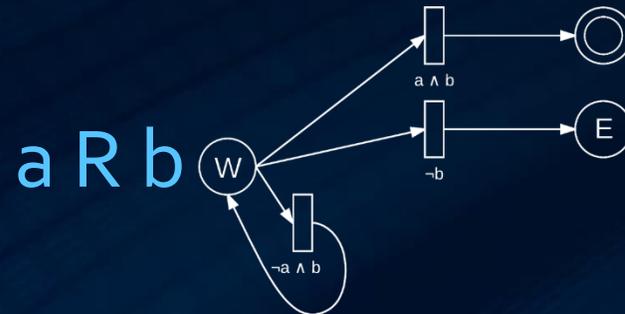
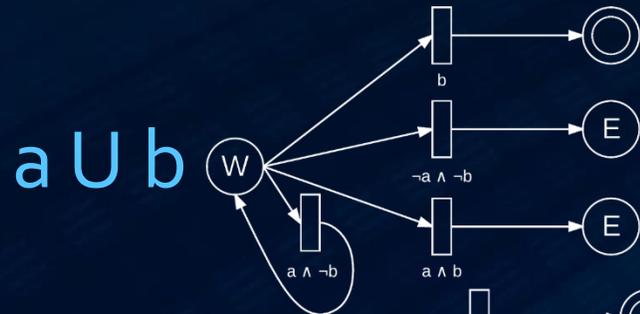
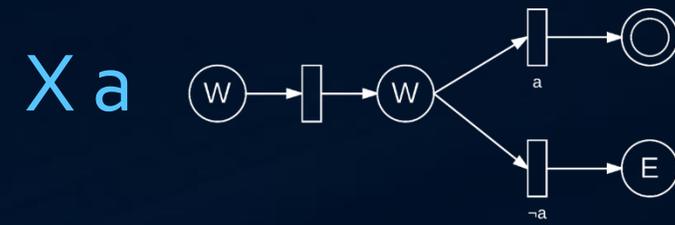
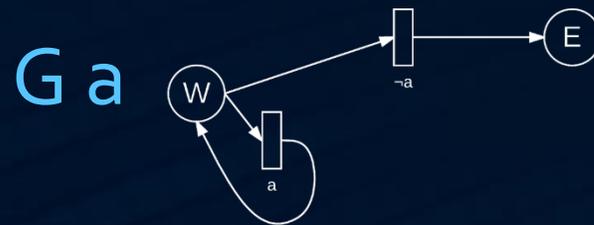
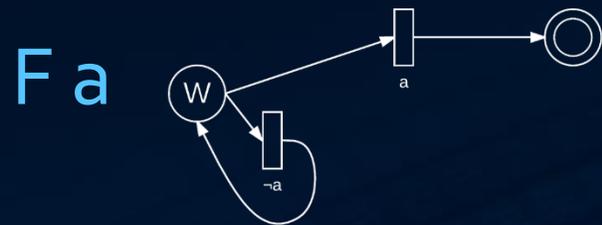
Modelli di Rete (1/2)



Ogni Formula Semplice, cioè formata solamente da un Operatore, è modellata da una Rete di Petri.

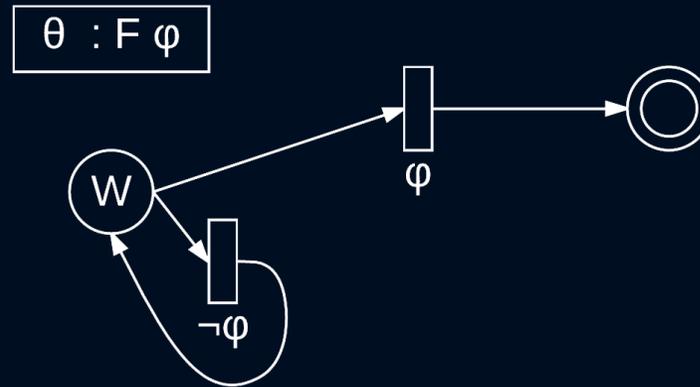
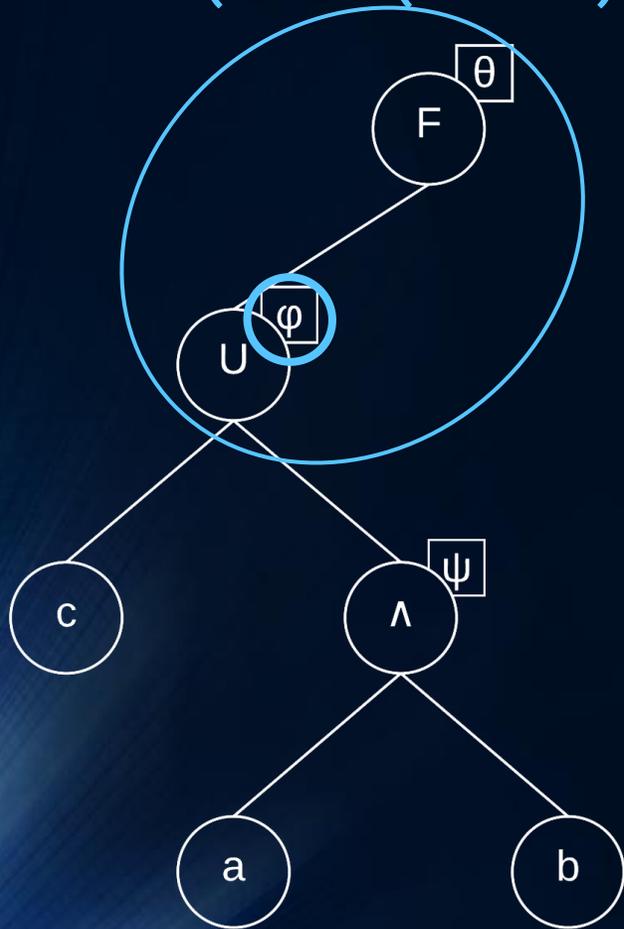
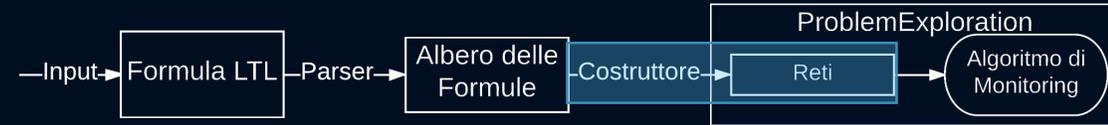


Modelli di Rete (2/2)



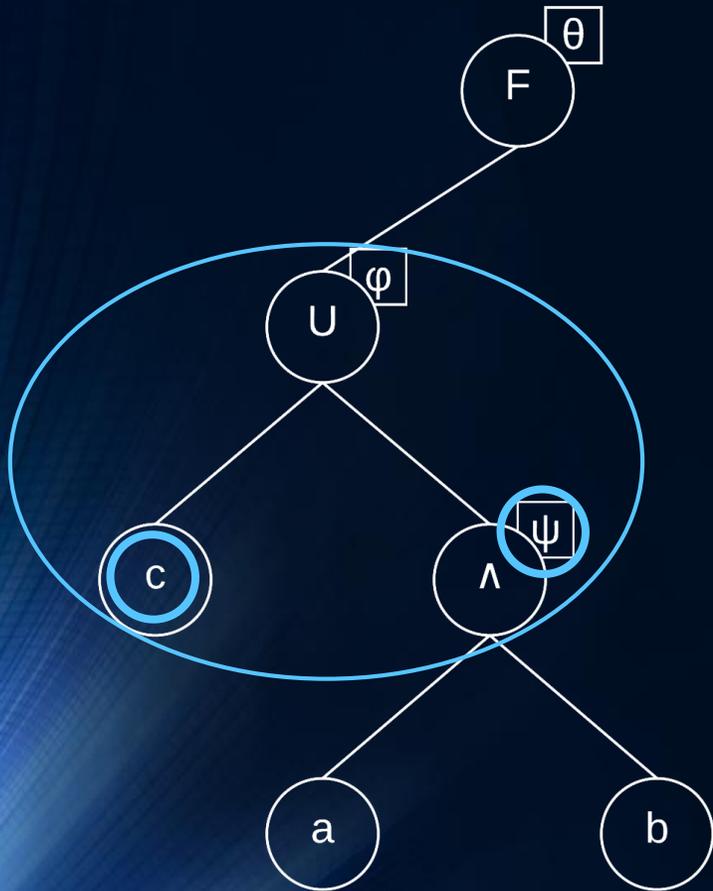
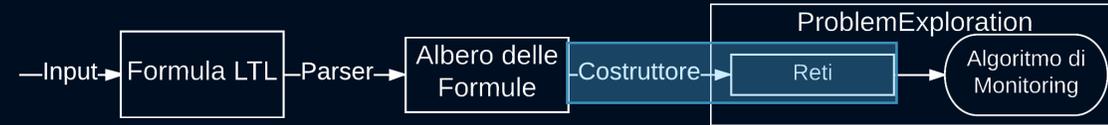
Costruzione delle Reti

$\theta : F(cU(a \wedge b))$

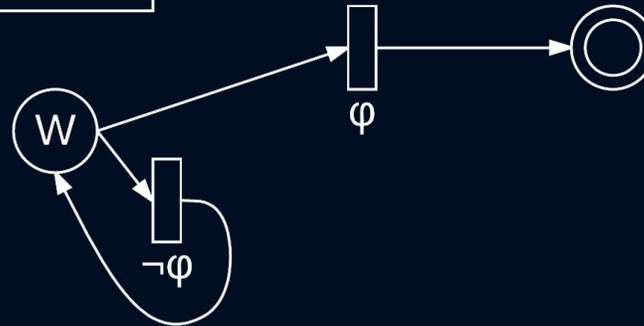


Costruzione delle Reti

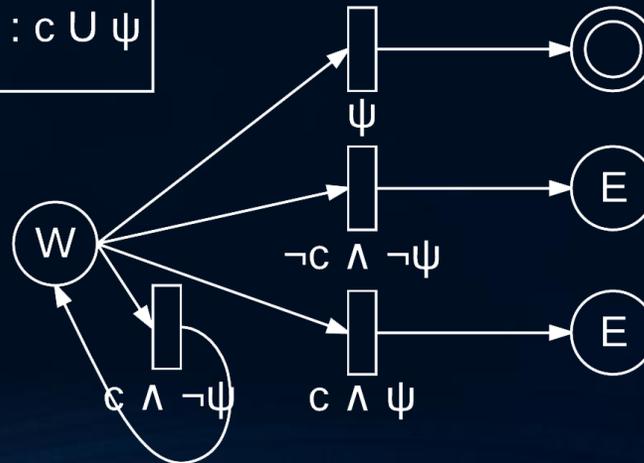
$\theta : F(cU(a \wedge b))$



$\theta : F \varphi$

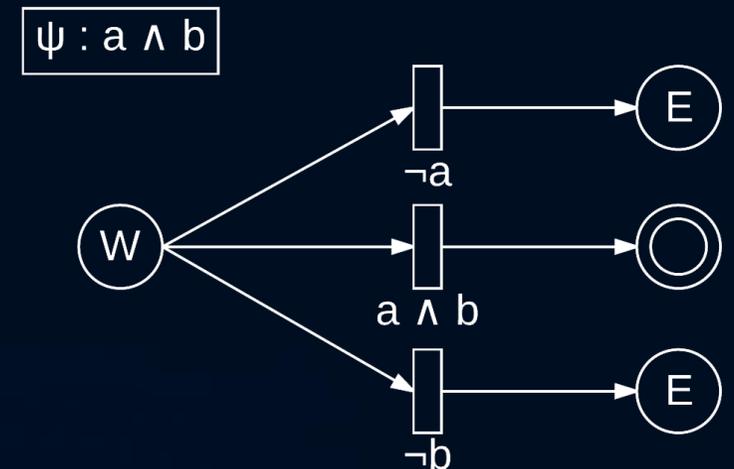
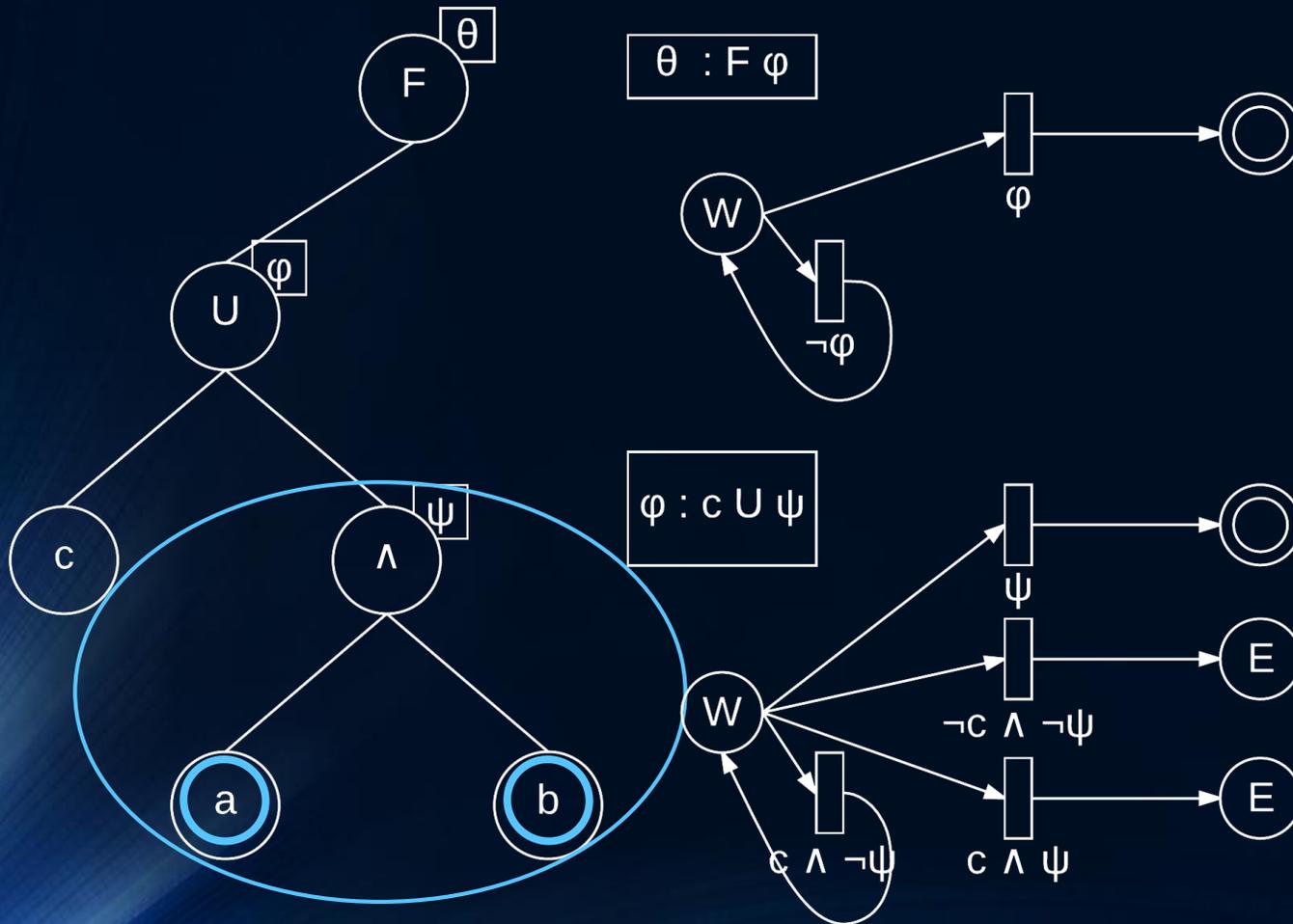
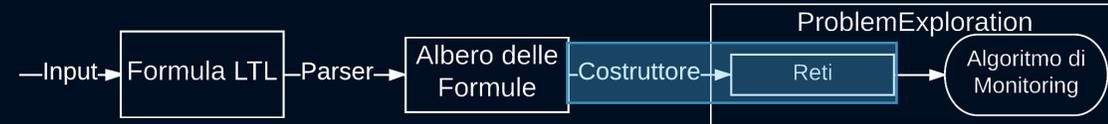


$\varphi : c U \psi$



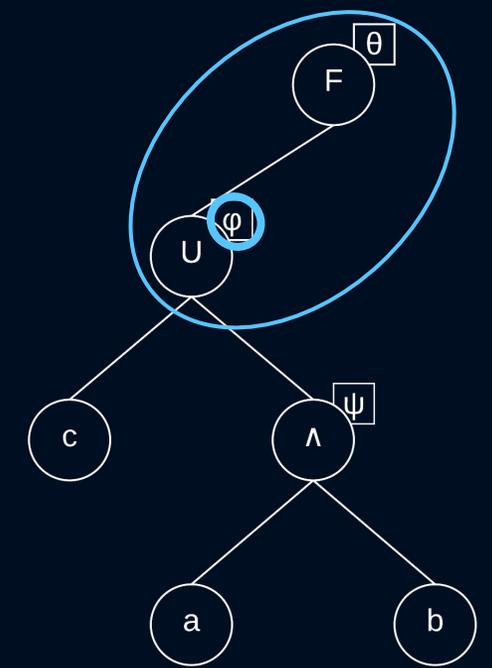
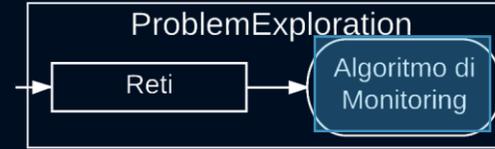
Costruzione delle Reti

$\theta : F(cU(a \wedge b))$

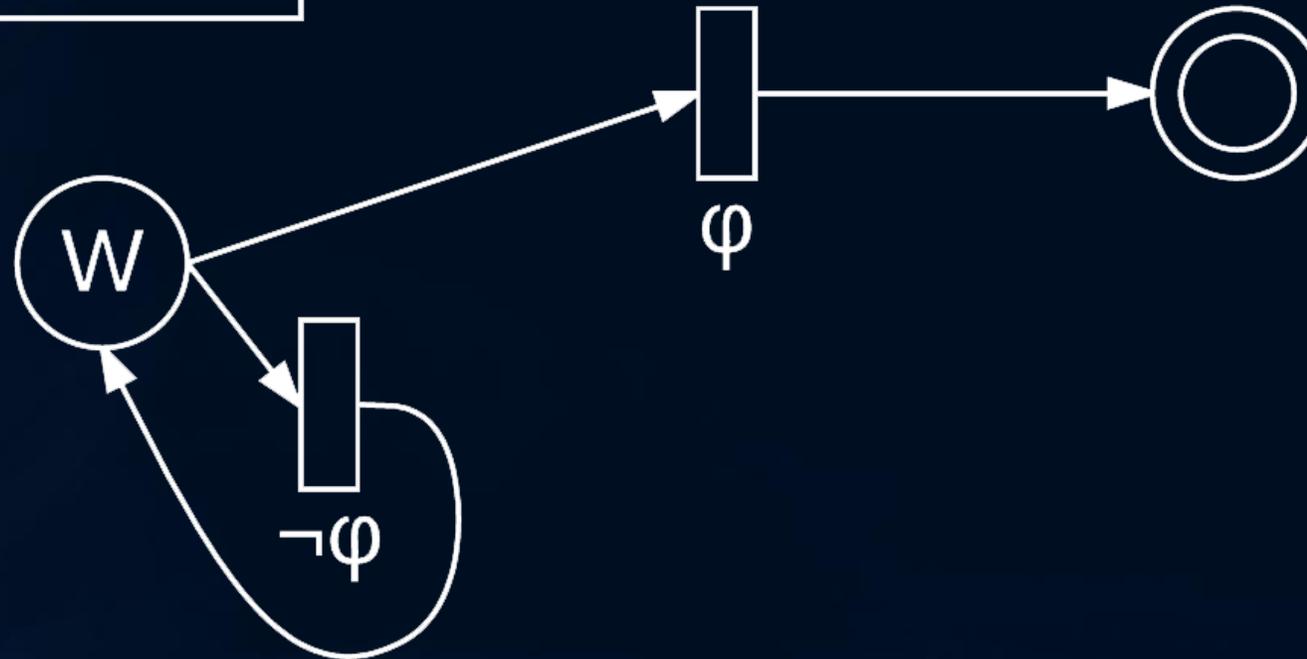


Funzionamento (1/4)

$\theta : F(cU(a \wedge b))$

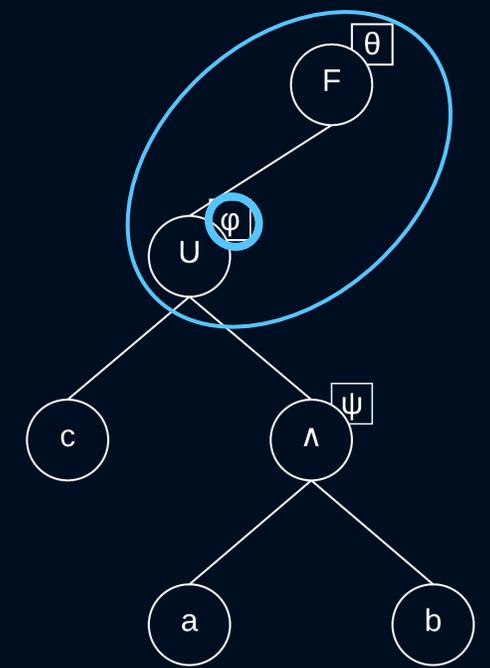
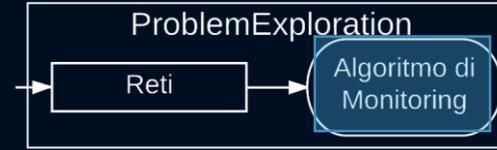


$\theta : F \varphi$

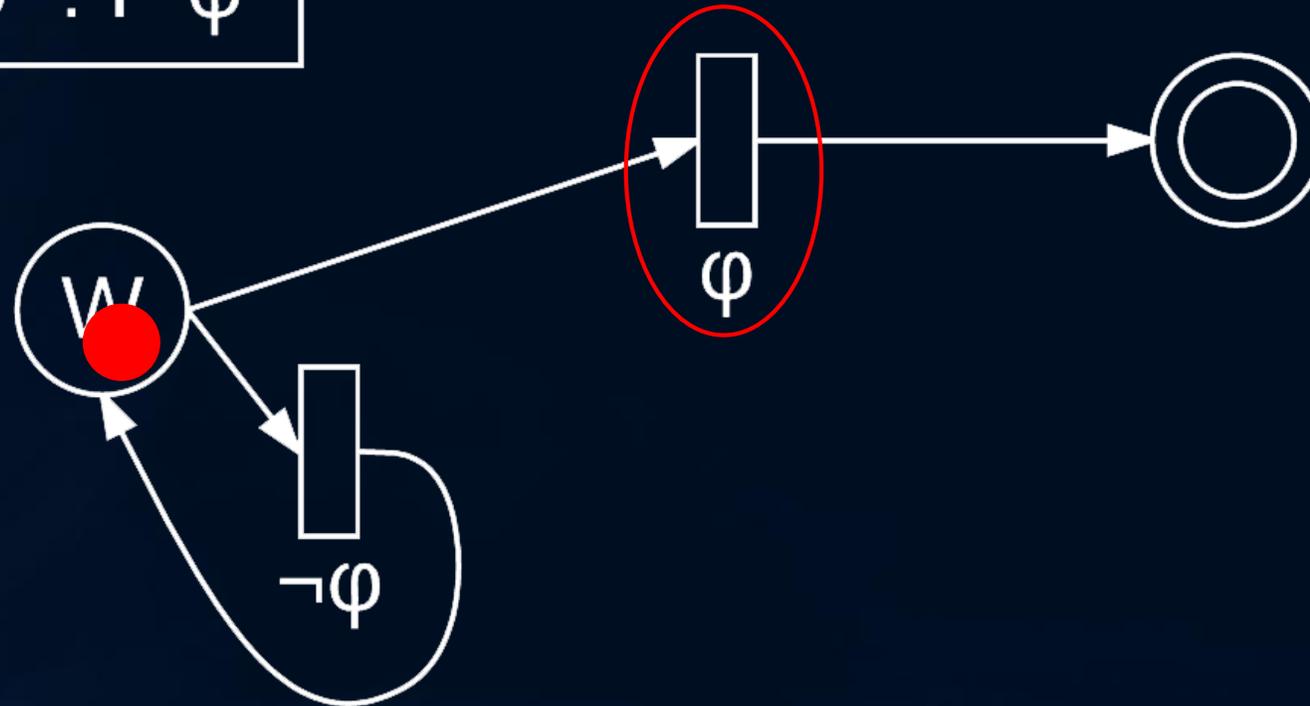


Funzionamento (1/4)

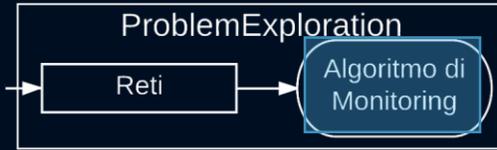
$\theta : F(cU(a \wedge b))$



$\theta : F \varphi$

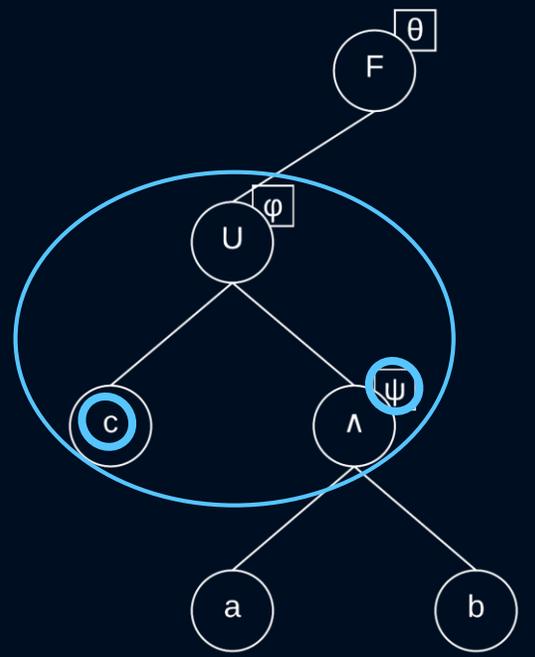
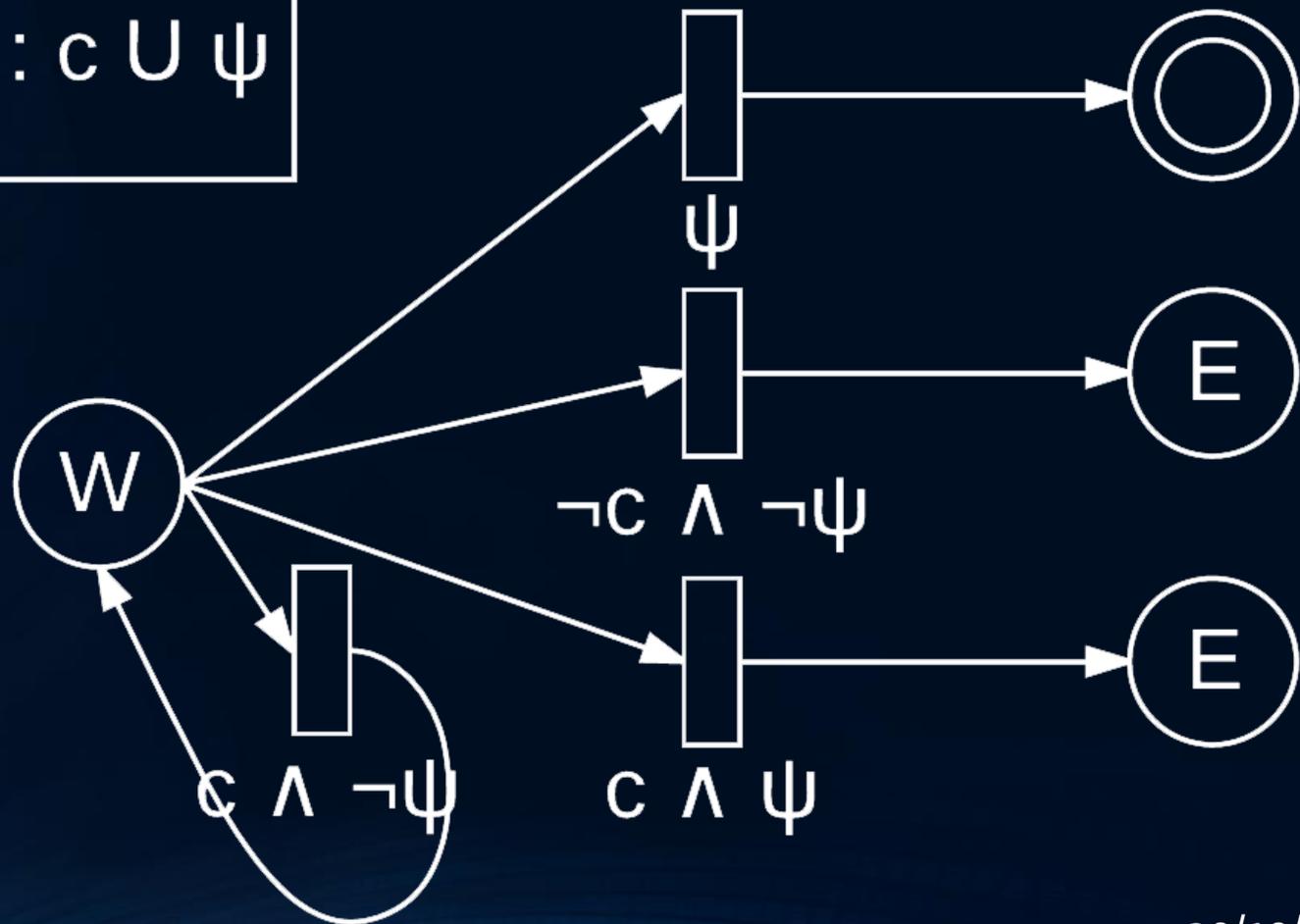


Funzionamento (2/4)

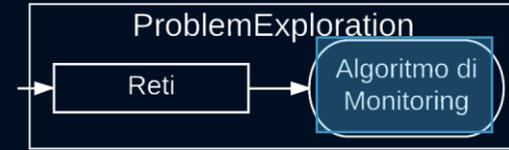


$\theta : F(cU(a \wedge b))$

$\varphi : cU\psi$

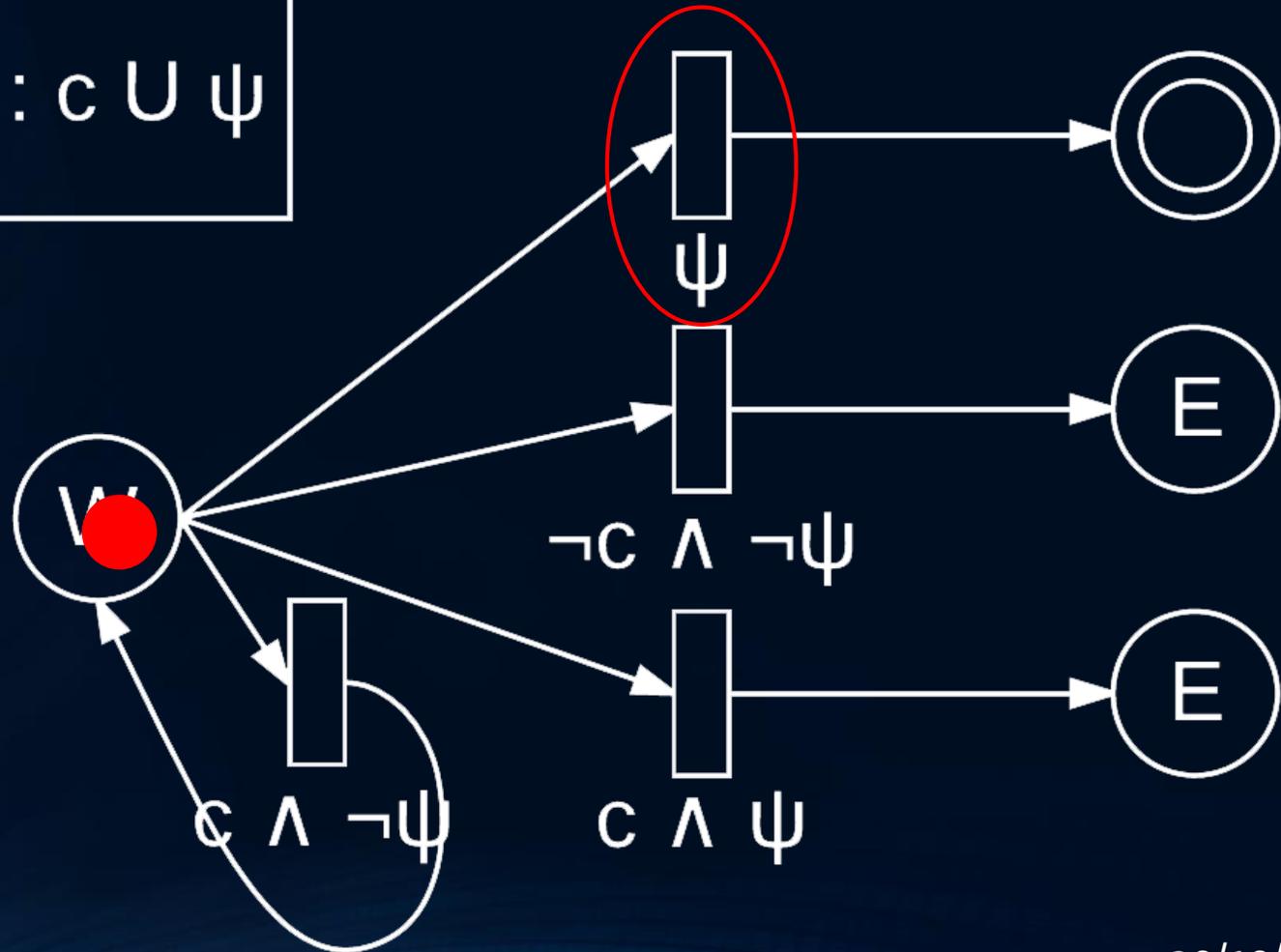
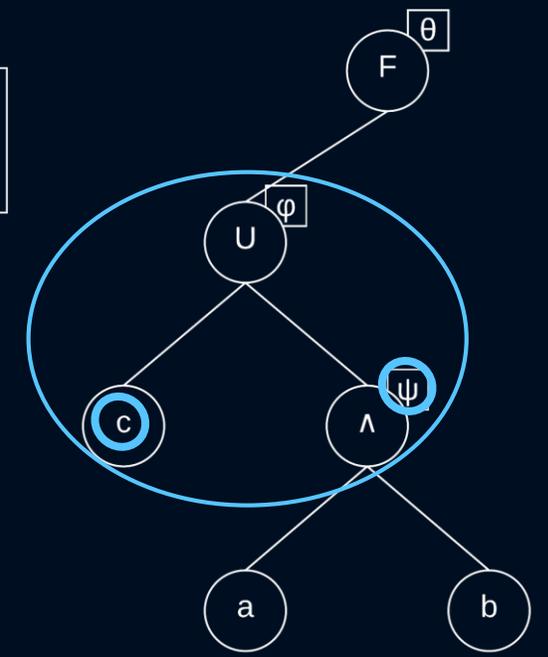


Funzionamento (2/4)

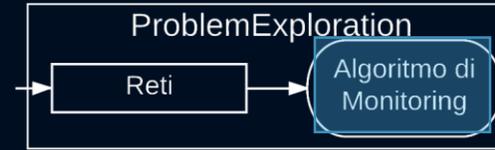


$\theta : F(cU(a \wedge b))$

$\varphi : cU\psi$

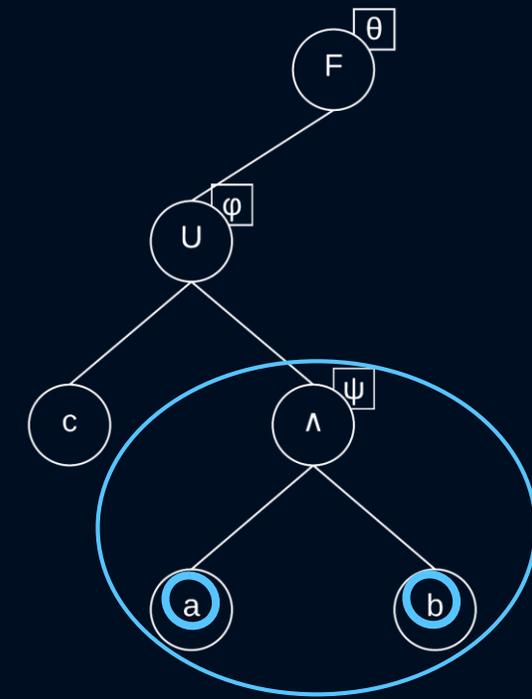
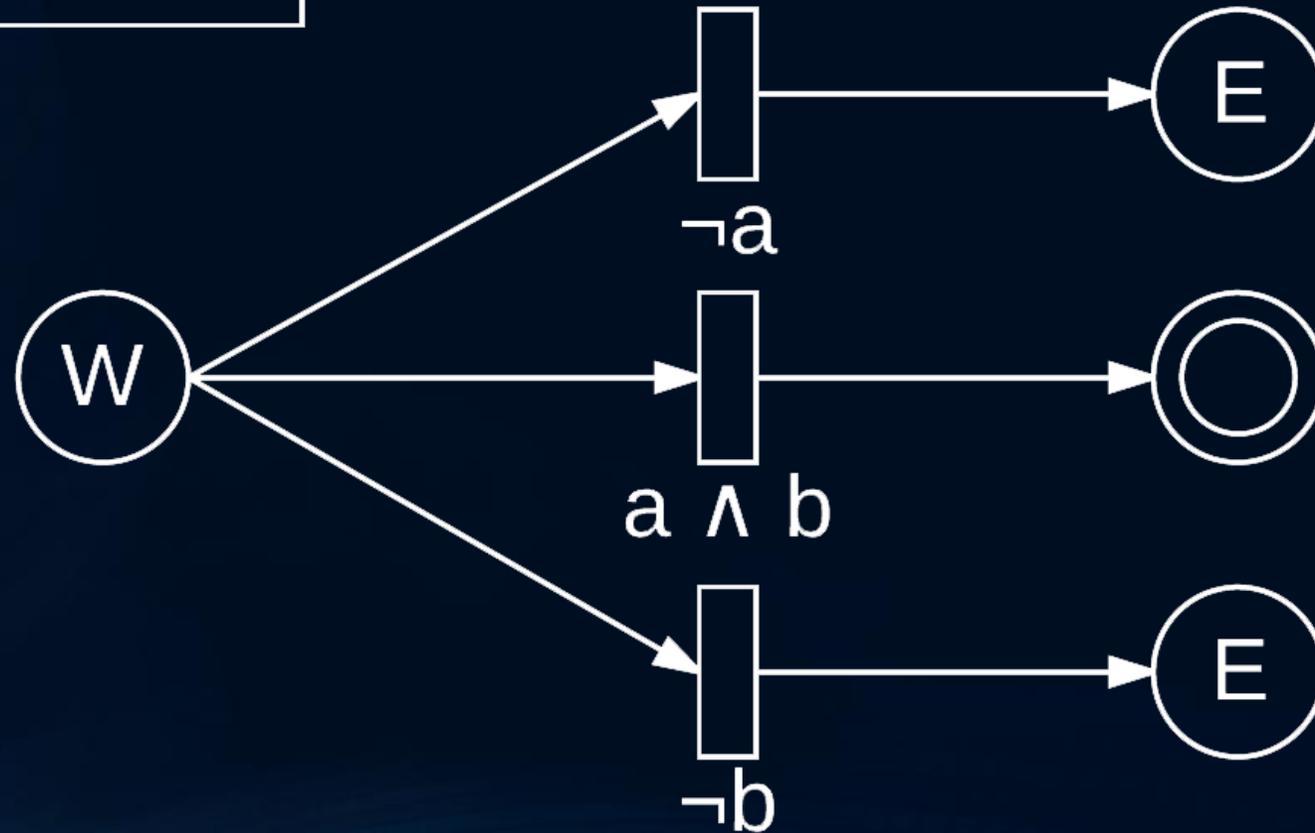


Funzionamento (3/4)

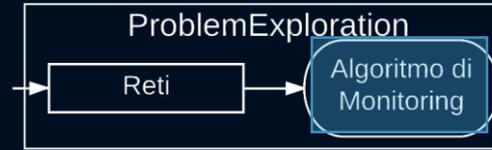


$\theta : F(cU(a \wedge b))$

$\psi : a \wedge b$

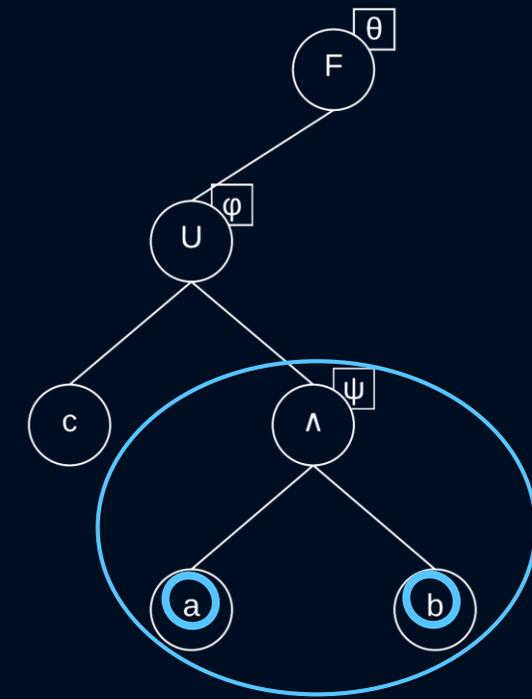
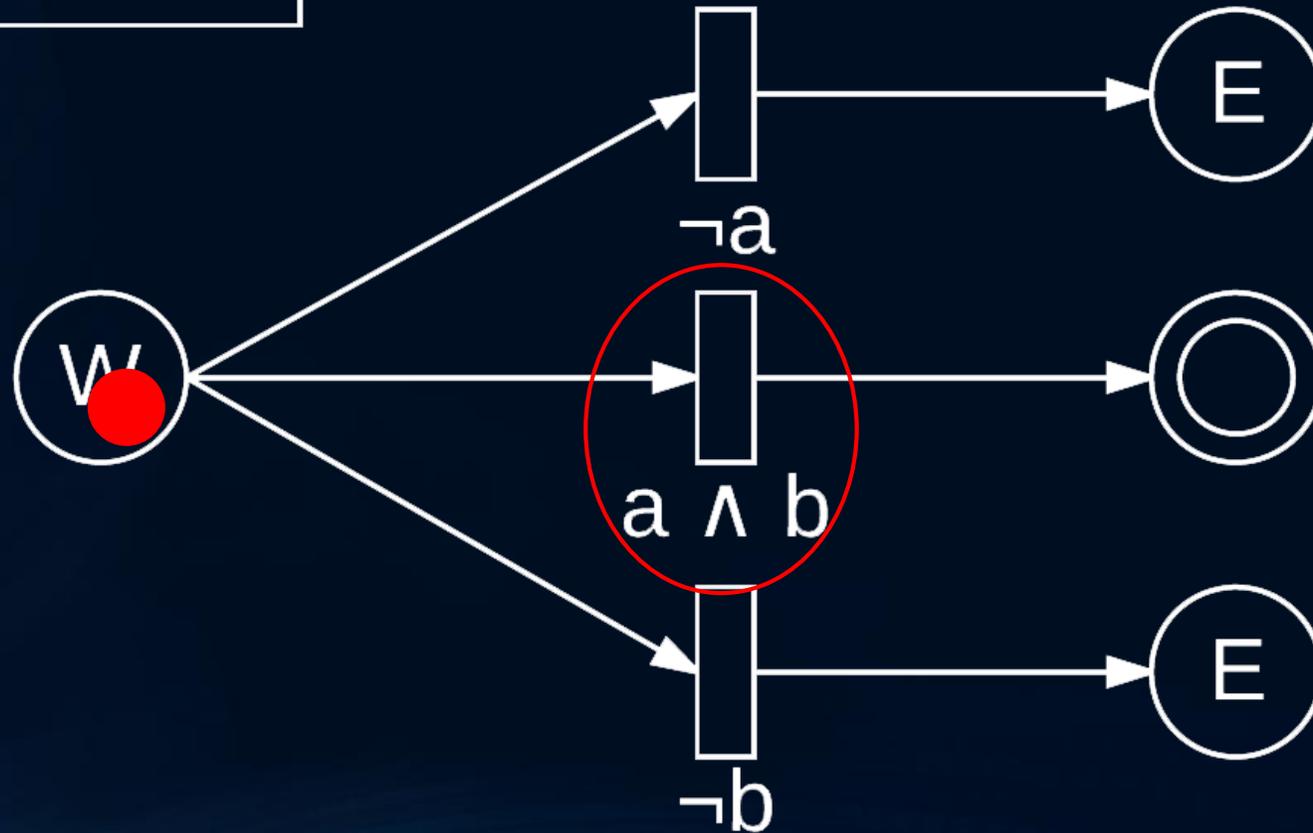


Funzionamento (3/4)



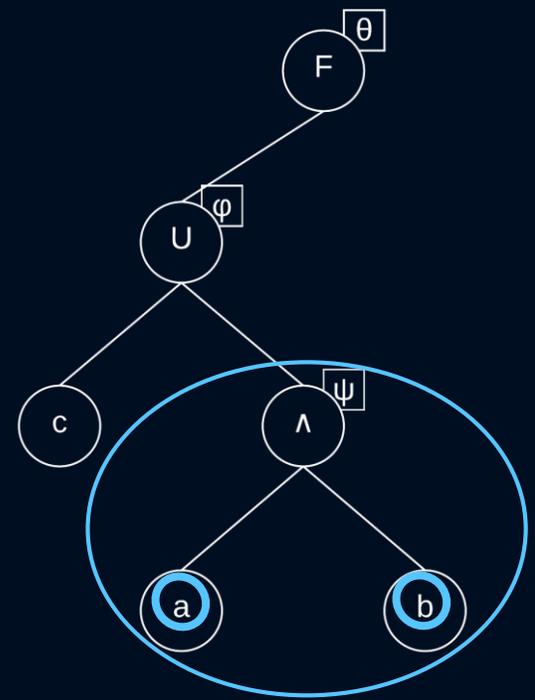
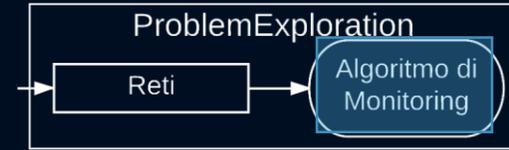
$\theta : F(cU(a \wedge b))$

$\psi : a \wedge b$

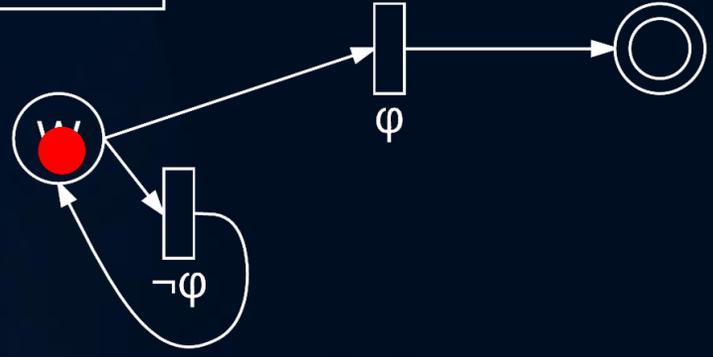


Funzionamento (4/4)

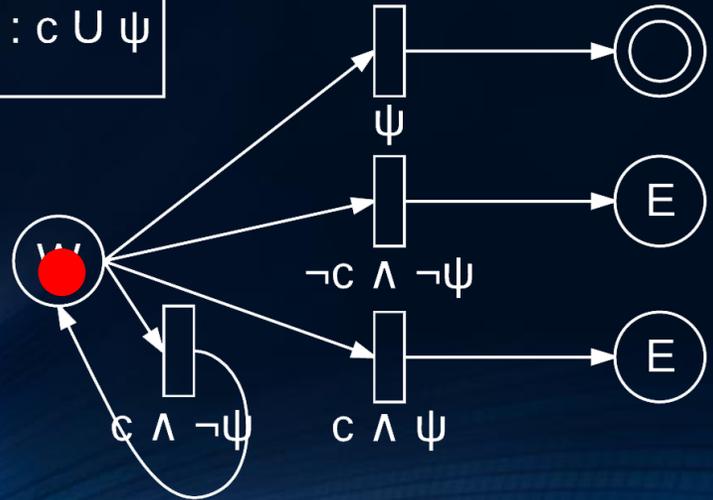
$\theta : F(cU(a \wedge b))$



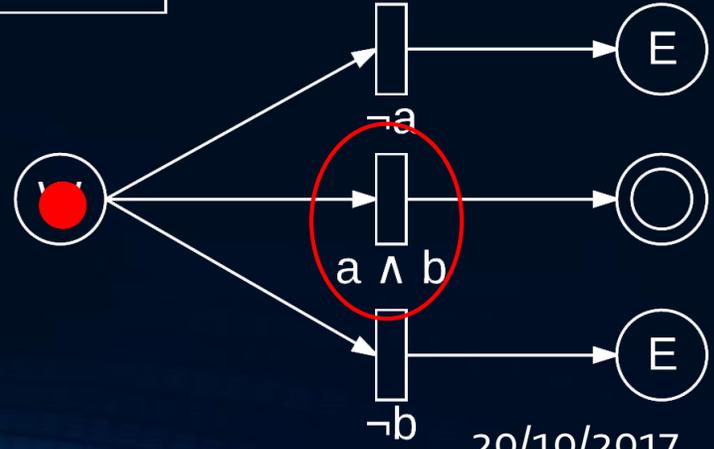
$\theta : F \varphi$



$\varphi : cU\psi$

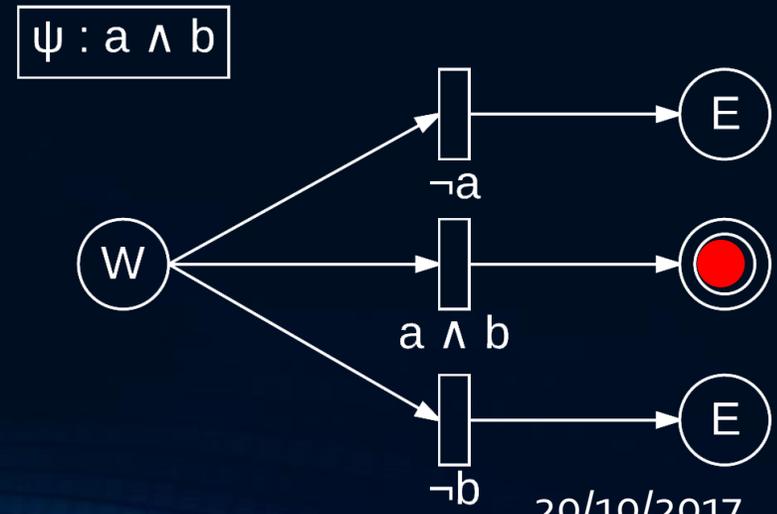
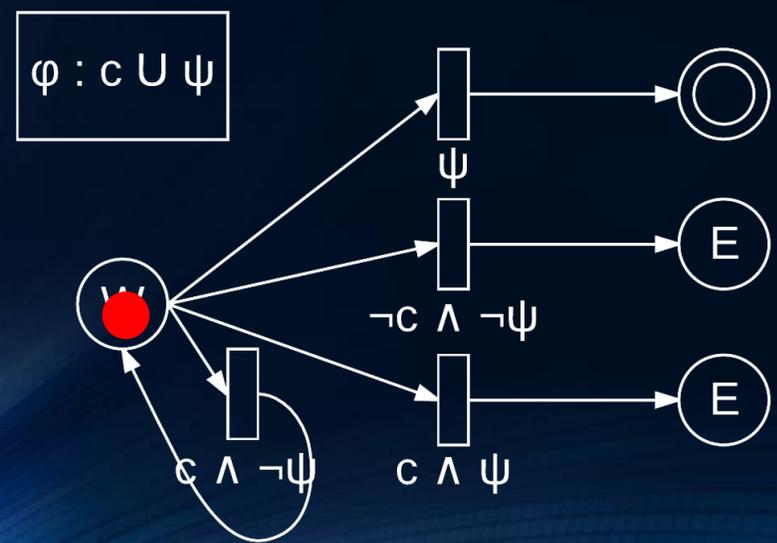
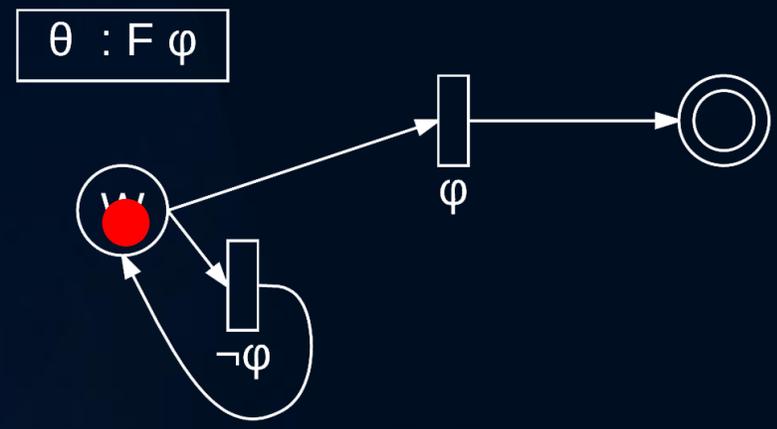
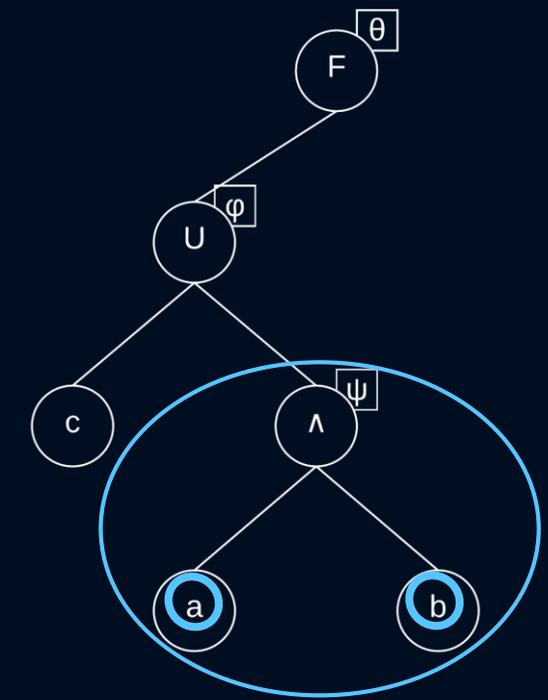
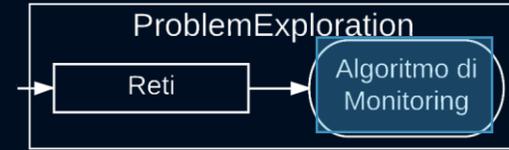


$\psi : a \wedge b$



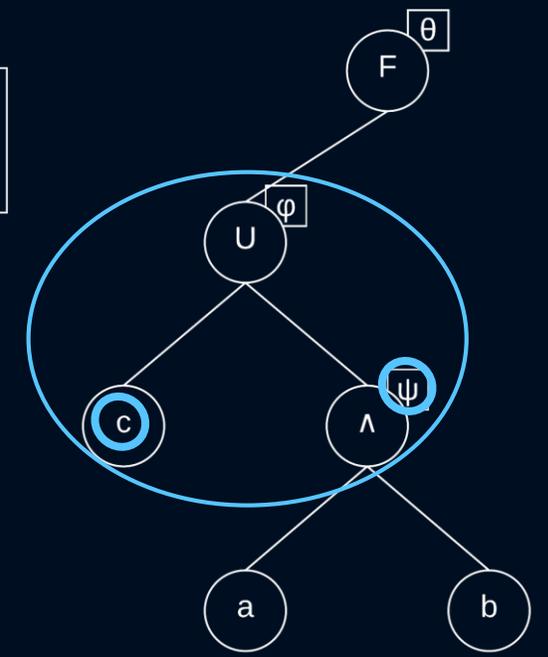
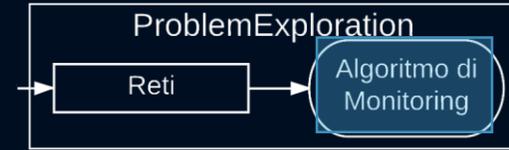
Funzionamento (4/4)

$\theta : F(cU(a \wedge b))$

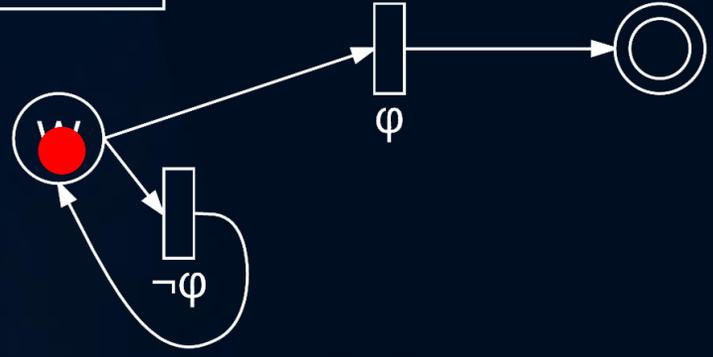


Funzionamento (4/4)

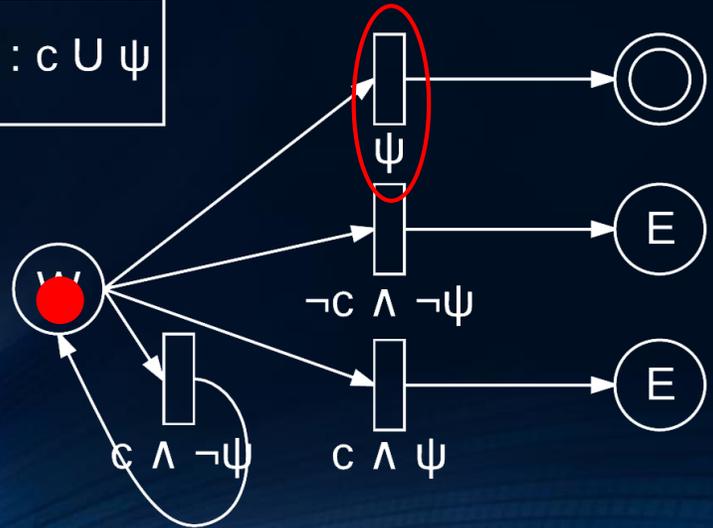
$\theta : F(cU(a \wedge b))$



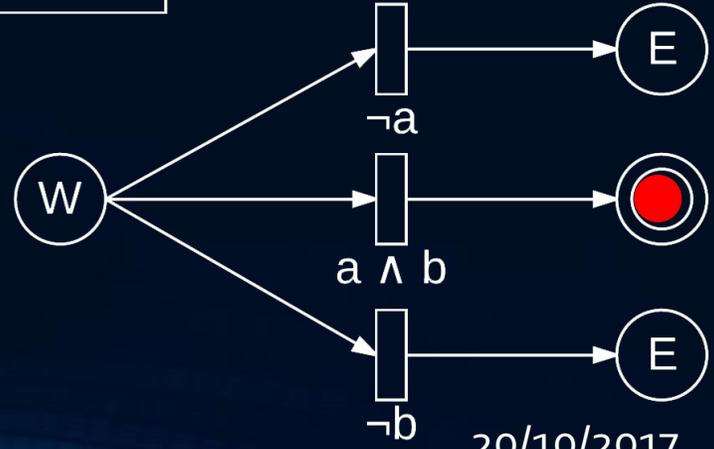
$\theta : F \phi$



$\phi : cU\psi$

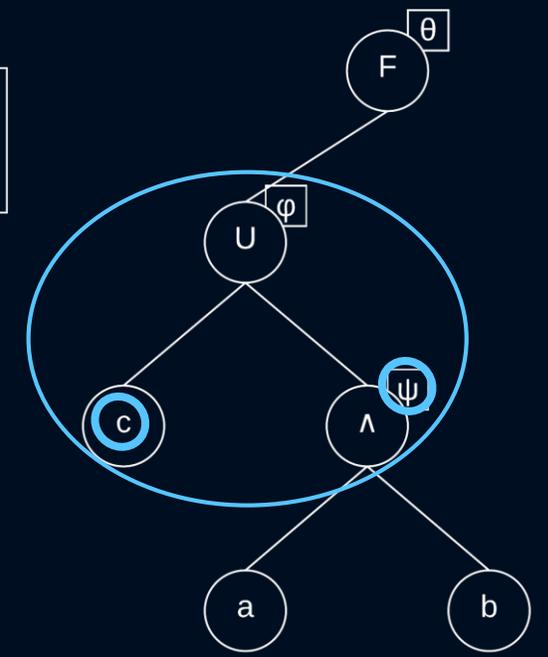
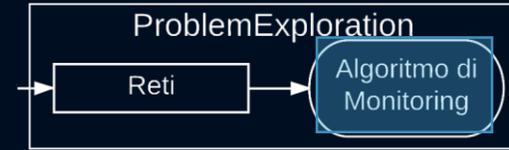


$\psi : a \wedge b$

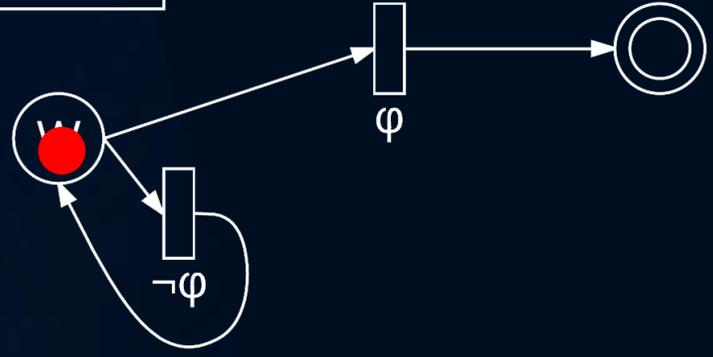


Funzionamento (4/4)

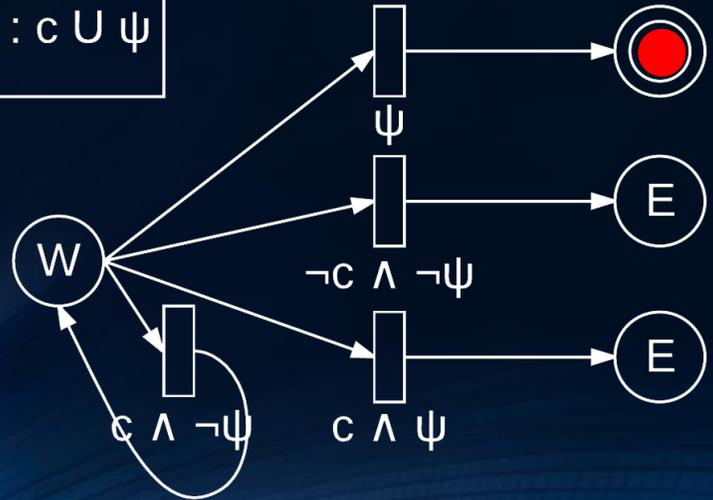
$\theta : F(cU(a \wedge b))$



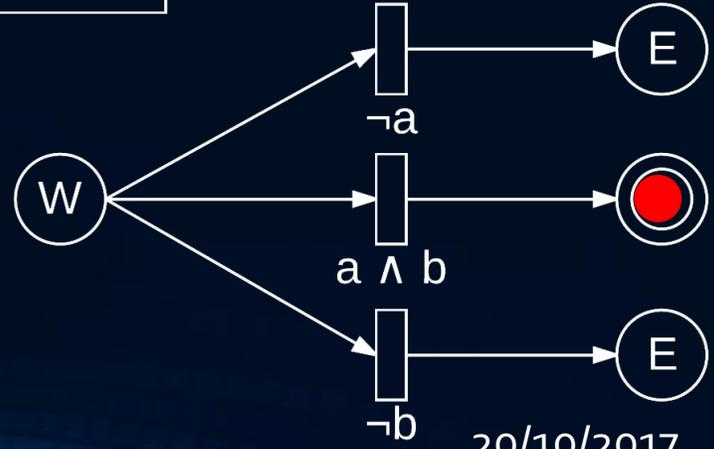
$\theta : F \varphi$



$\varphi : cU\psi$

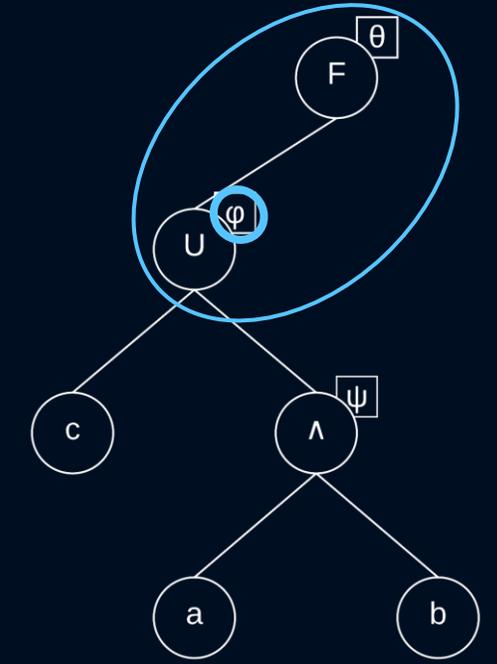
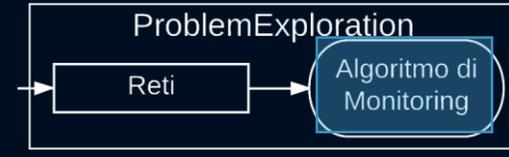


$\psi : a \wedge b$

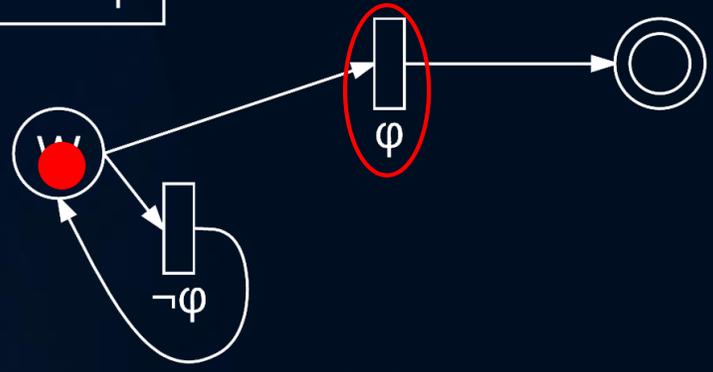


Funzionamento (4/4)

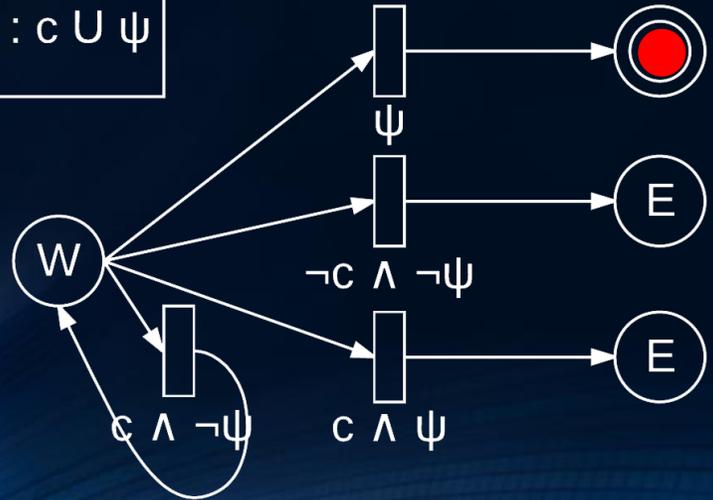
$\theta : F(cU(a \wedge b))$



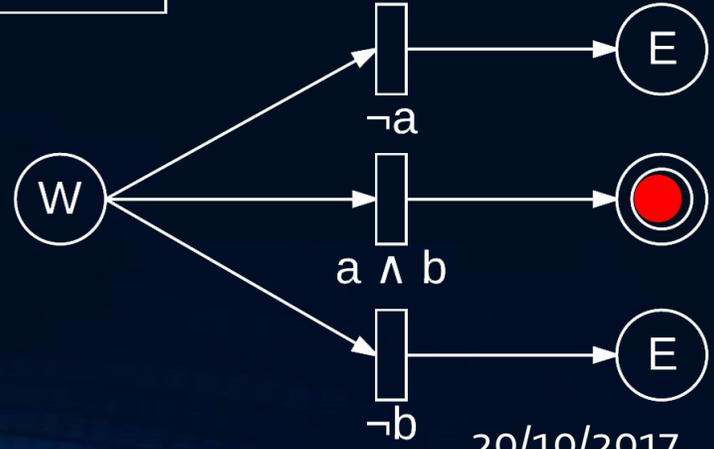
$\theta : F \varphi$



$\varphi : cU\psi$

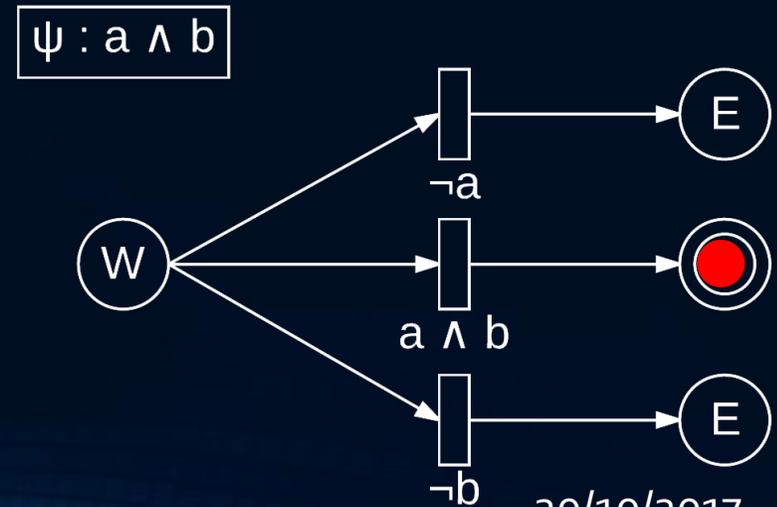
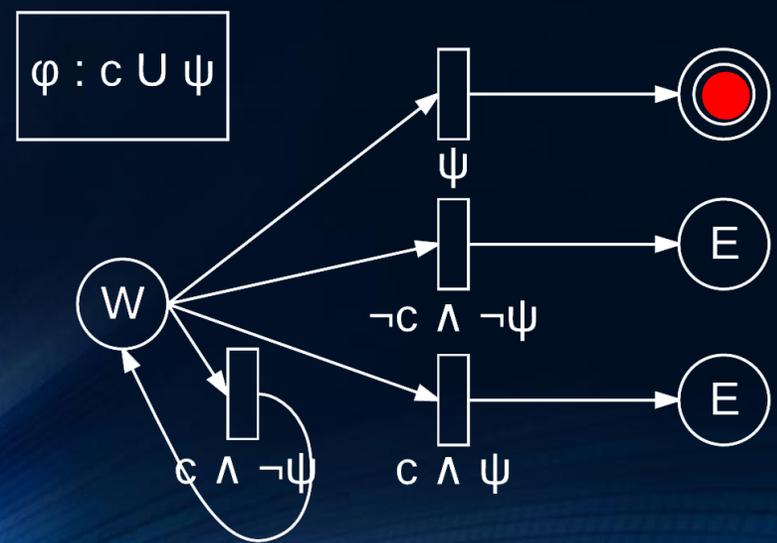
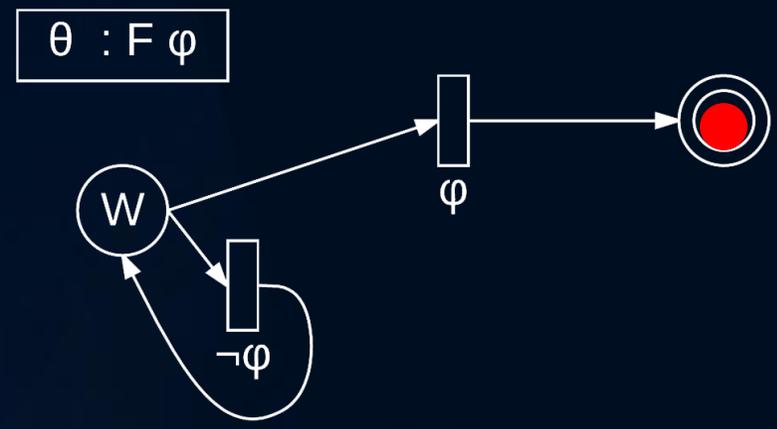
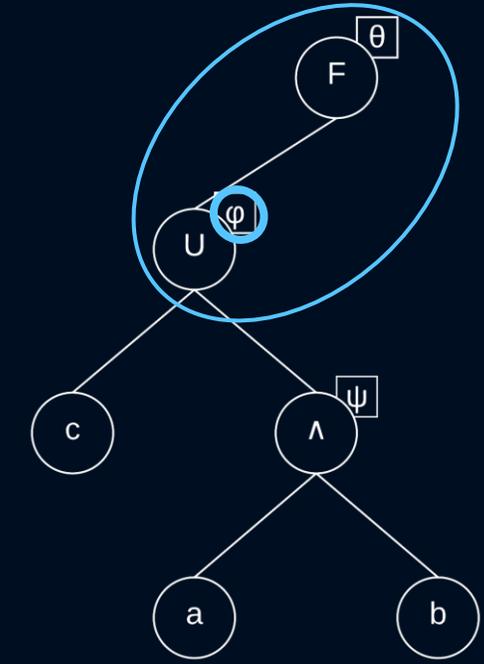
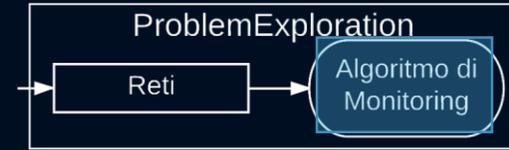


$\psi : a \wedge b$



Funzionamento (4/4)

$\theta : F(cU(a \wedge b))$



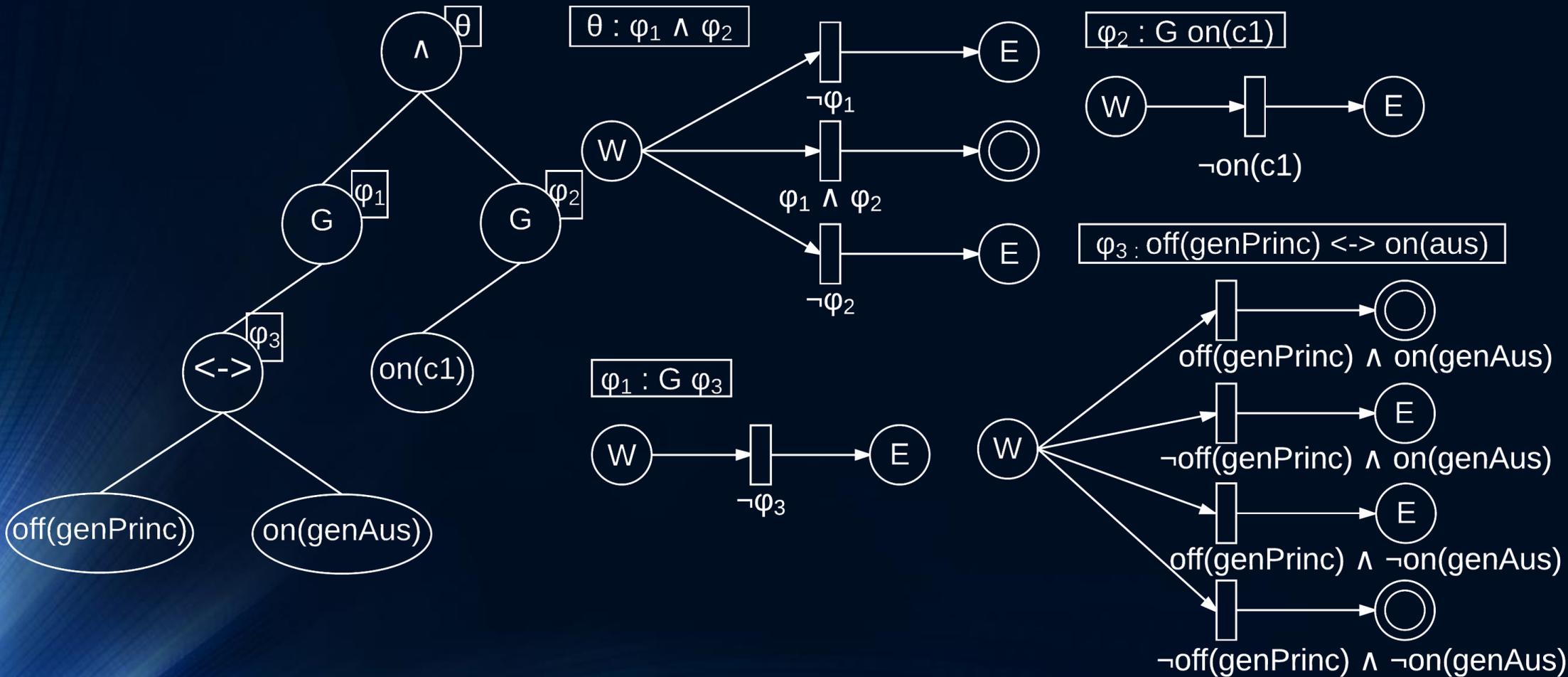
Validazione

Goals

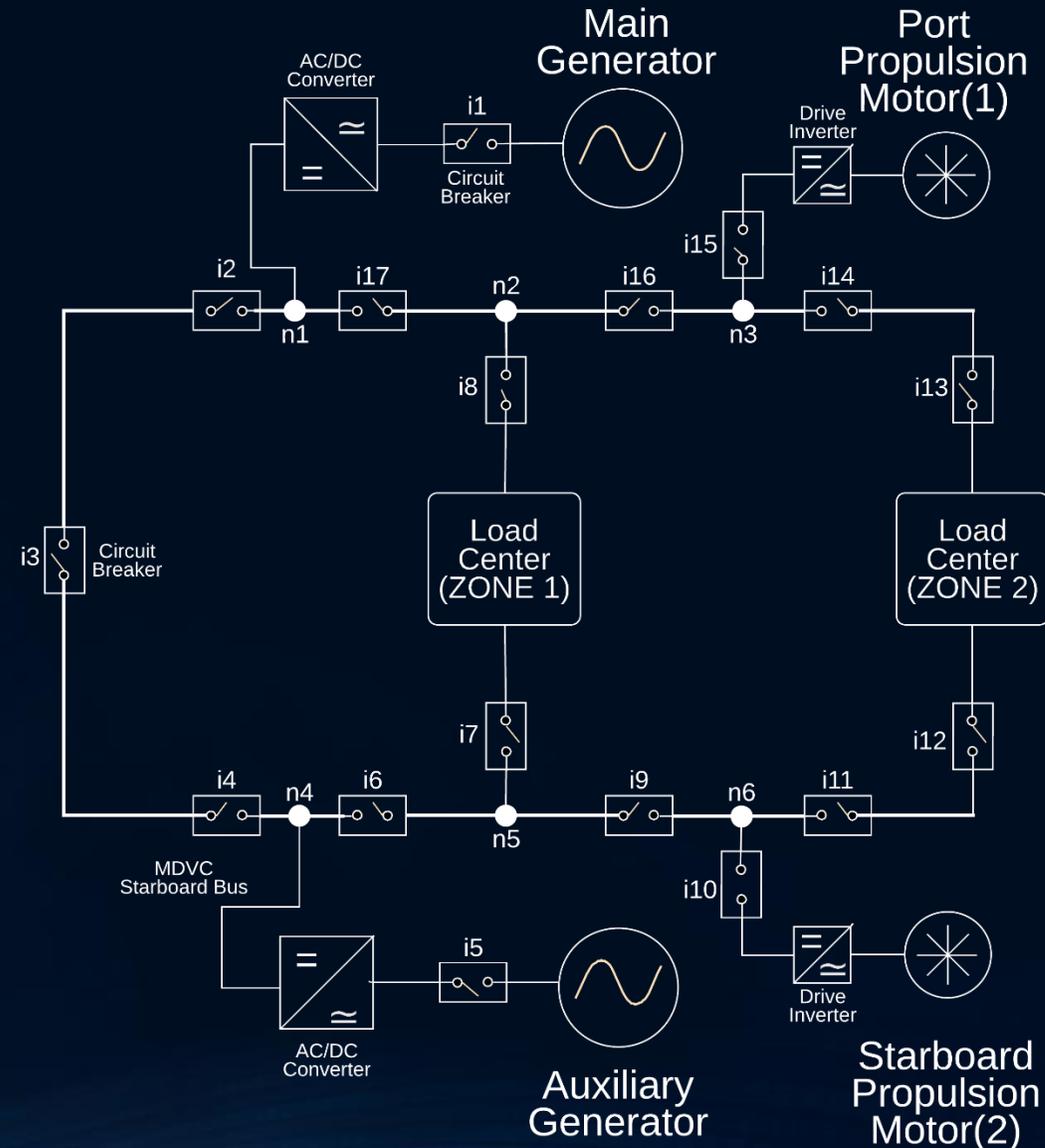
- Se il Generatore Principale si guasta bisogna accendere il Generatore Ausiliario
 - Il Carico 1 deve essere sempre alimentato
 - Il Carico 2 deve essere alimentato finché è acceso il Motore 2
 - Se si verifica un incendio bisogna staccare l'alimentazione al Carico 2 e accendere il sistema antincendio
- $G (\text{off}(\text{genPrin}) \leftrightarrow \text{on}(\text{genAus}))$
 - $G (\text{on}(c1))$
 - $G (\text{on}(c2) U \text{off}(m2))$
 - $G (\text{verified}(\text{inc}) \rightarrow (X \text{off}(c2) \wedge F \text{on}(\text{antiInc})))$

Validazione - Monitor

$$G(\text{off}(\text{genPrin}) \leftrightarrow \text{on}(\text{genAus})) \wedge G(\text{on}(\text{c1}))$$

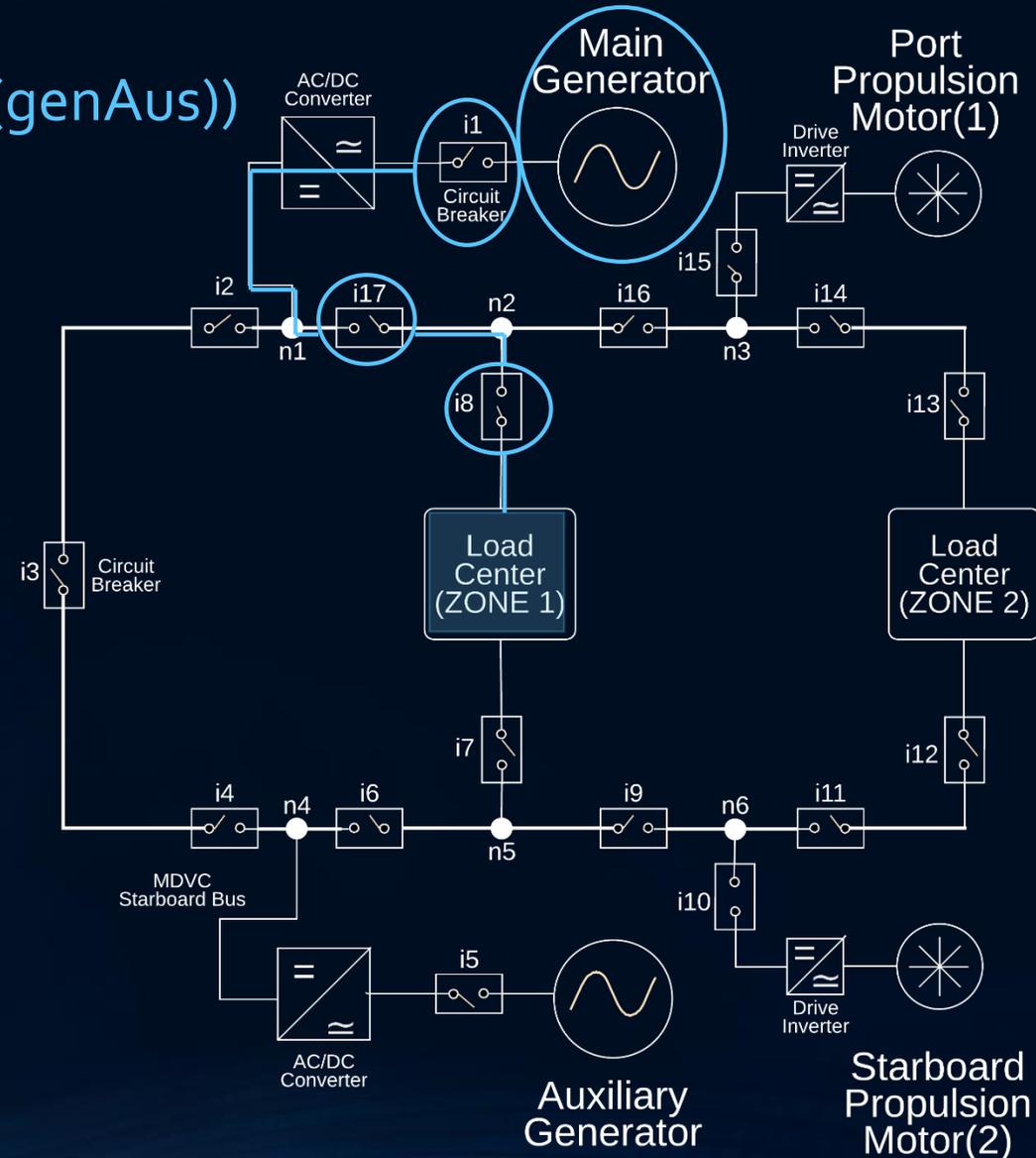


Validazione - Sistema di Alimentazione (1/3)



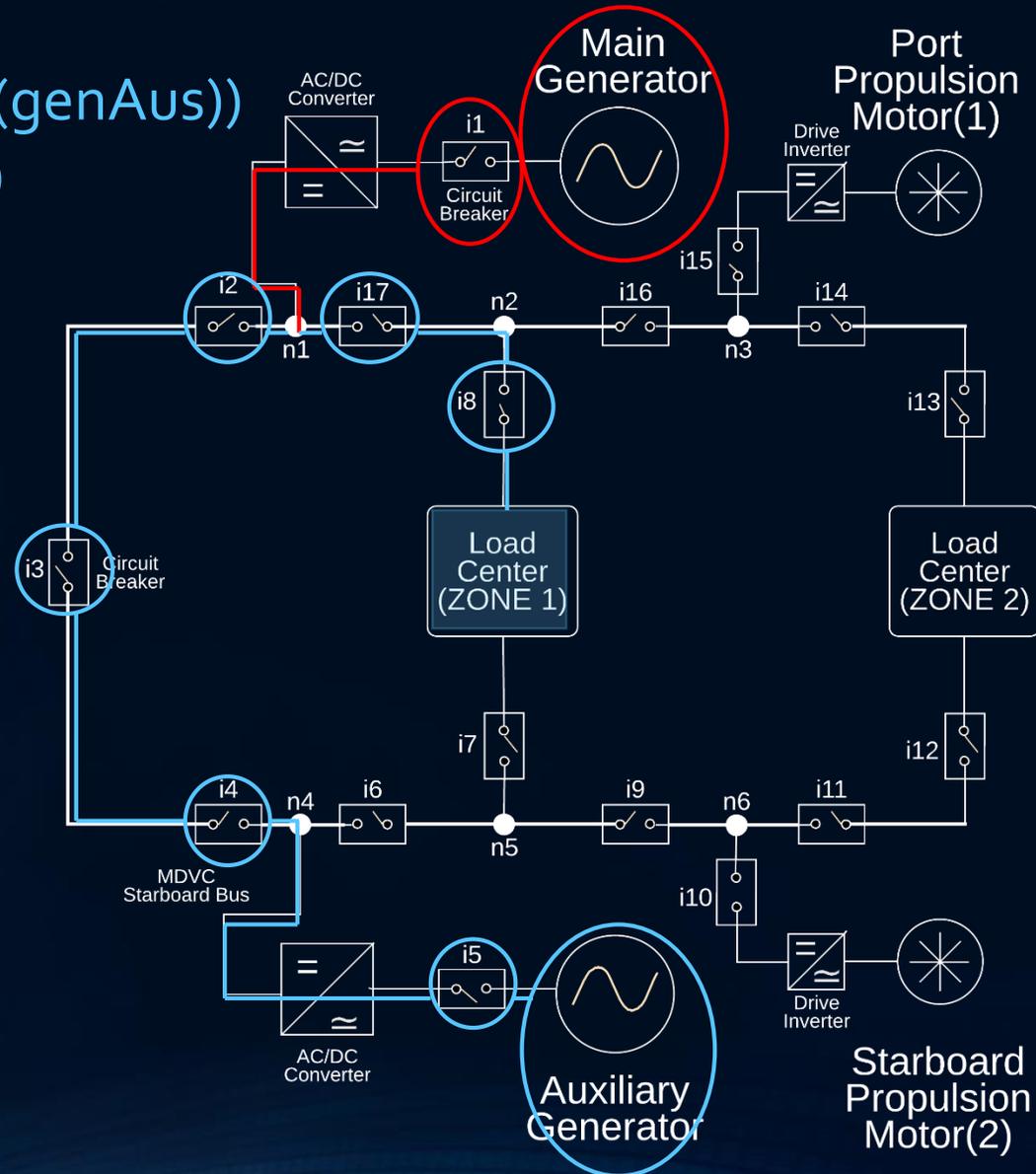
Validazione - Sistema di Alimentazione (2/3)

$G(\text{off}(\text{genPrin}) \leftrightarrow \text{on}(\text{genAus}))$
 $\wedge G(\text{on}(\text{c1}))$



Validazione - Sistema di Alimentazione (3/3)

$G(\text{off}(\text{genPrin}) \leftrightarrow \text{on}(\text{genAus}))$
 $\wedge G(\text{on}(c1))$



Capabilities:

- $\text{on}(\text{auxiliary})$
- $\text{close}(i5)$
- $\text{close}(i4)$
- $\text{close}(i3)$
- $\text{close}(i2)$

Conclusioni (1/2)

- Maggiore Espressività

Tramite l'utilizzo degli Operatori Temporalis viene estesa la potenza del linguaggio per i Goals e quindi più possibilità per la specifica dei Requisiti.

- Confronto con il Model Checking

- Dato il modello di un sistema, con il Model Checking la verifica di una Formula LTL avviene tramite la costruzione di un Automa.
- L'utilizzo di una struttura basata su Rete di Petri invece degli Automi, porta benefici in termini di complessità della struttura e tempi di costruzione.

Conclusioni (2/2)

- **Sviluppi Futuri**
 - Questo lavoro ha messo le basi per la costruzione di un'unica Rete di Petri per il Monitoring che potrebbe portare ad ulteriori benefici.
 - È in preparazione l'articolo scientifico per l' International Conference on Agents and Artificial Intelligence (ICAART) 2018, sessione ASAMA <https://icaart.org/ASAMA.aspx>
- **MUSA 2.0**

Disponibile open source su https://github.com/icar-aose/musa_2/