UNIVERSITÀ DEGLI STUDI DI MILANO

LAUREA MAGISTRALE
Dr. EMANUELE MERONI

[MODELLAZIONE ANALISI SISTEMI – TEORIA]

ASM ABSTRACT STATE MACHINE

PARALLELISMO = non sequenzialità
SUPERUNIVERSO = associazione SIMBOLI e SIGNIFICATO

SCRITTURA IN SEQ. DEL CODICE

1. HEADER: **SIGNATURE** (dichiarazioni)
2. BODY: **DEFINITIONS (limiti)**
3. **MAIN RULE** (inizio programma)
4. **DEFAULT INIT S0** (inizializzazione)

FUNZIONI DI DEFAULT IN ASM

NOT() opposto boolean es lele:=false NOT(lele) lele diventa true

PAROLE SPECIFICHE ASM

- Significa che la funziona ritorna valore (**il return è omesso**)

TIPI: Boolean, Integer String …
function si usa per usare una variabile non var
r_main = (è un'associazione al main non come di solito che contiene)
MONITORED "monitorata" in INPUT
CONTROLLED var interna
DYNAMIC può variare nel corso del programma
DERIVED torna il tipo dichiarato es: derived F: Integer **-> Integer**
$ definizione variabile es **$x in** Integer
associazione al tipo dynamic controlled lele: Integer
MACRO RULE si usa per definire una funzione
r_ nomenclatura per distingere le funzioni (regole) es **r**_calcola[]
choose usata per la randomicità
FUNZIONE **0-ARIA** = attributo (arietà)
LET assegnamento costanti
ASM nomeFile
Import ./StandardLibrary (dà la semantica al codice)

UNIVERSITÀ DEGLI STUDI DI MILANO

LAUREA MAGISTRALE
Dr. EMANUELE MERONI

SINTASSI ASM

Ricordati che nell'IF puoi mettere solo variabili che hai già definito. Classico errore **dell'ENUM aggiornato.**
LA LUNGHEZZA DI OGNI PAROLA DELL'ENUM DEVE ESSERE ALMENO 1
CONFRONTO ENUM SENZA ""

SINTASSI MOLTO PARTICOLARE: DENTRO IL IF CI PUO ESSERE UN COSTRUTTO TIPO SWITCH MA NON ANCHE UNA ASSEGNAZIONE SE NELLO SWITCH
NON C'E' IL ;
LE VARIABILI IL PRIMO CARATTERE SEMPRE MINUSCOLO
NON USARE LA VARIABILE 'i' E' PAROLA CHIAVE, USA J PER L'INDICE
CONTATORI Naturali 'n' esempio function i = 0n (SENZA due punti)
INPUT NON VISIBILE DAL CODICE (preciso istante, sai solo che c'è)
choose $s in Sign **with true** continua a scegliere (partita)
assegnamento,**aggiornamento di valore : =** es a = b errore a : = b ok

CHIAMATE [ ] non con la ()	r_funzione[ ]
domain: creazione tipo	domain lele subsetof Integer
range:	domain lele **= {0..13}**
enum casistiche	**enum** domain lele = {maschio | femmina}
Matrice	controlled matrice:

NO graffe/no break-switch, **no WHILE**
si tonde al CASE-switch
NO "default" nello switch (**otherwise che NON ha i :**)
(forall, skip, seq,par,exist,then,endswitch,endIf

TIPS: Dato che è pieno di BUG!! Ti do dei consigli:

"CHIUDI SEMPRE GLI ALTRI sorgenti" SE STAI TESTANDO 1 FILE APRI SOLO QUELLO.
(Ho premuto 3 e mi dato l'output di un'altro mio file aperto che infatti avevo definito il caso 3!!)

SE NON SI AGGIORNA INSTANTANEAMENTE RIPARSA / chiudi riapri ECLIPSE

NON COMMENTARE NEL DEFAULT INIT altrimenti ti dà errore

DEFAULT INIT SO NON OBBLIGATORIO METTERLO SE DEVI INIZIALIZZARE

SE CREI UN ENUM SEI COSTRETTO A USARLO NEL MAIN E LA LUNGHEZZA > 1 DELLE PAROLE!!

LOGICA ASM

DEFAULT INIT S0: da i valori alle variabili

- input è automatico se dichiari una monitored (function controlled = monitored)
- CONTATORI 'n' (**function** indice = 0**n**)
- A = 0 SOLO NEL MAIN SI METTE IL := PERCHE NELLA SINTASSI NON ESISTE UN ==

MAIN RULE: viene chiamato in loop (a:=a+1 nel main)
SE NON METTI UNA CONDIZIONE DENTRO IL MAIN: lo eseguirà random volte
SE LA METTI ESEGUIRA' IL MAIN SECONDO LA CONDIZIONE/i (TRUE)

Per es: if (indice < N) **then par …** endpar endif

➔ Da cui si deduce che l'utente NON PUO' DECIDERE QUANTI VALORI INSERIRE PERCHE' SE METTI INPUT N NEL MAIN AD OGNI CICLO VERRA RIDEFINITO

COME RAGIONA IL COMPILATORE

PERCHE A VOLTE CHIAMA IL MAIN ("WHILE-TRUE") E A VOLTE RANDOM?

Aaaaah. SE <MONITORATA> E' TUTTO NORMALE (eseguito 1 o 2 volte il main)
SE <CONTROLLED> ALLORA SI: IL MAIN E' COME SE FOSSE UN CICLO

1 istruzione ALLA VOLTA, NE VUOI DI PIU? USA PAR (NON CI SONO LE GRAFFE)

Le singole instanze di istruzioni che vengono eseguite (e.g: output / assegnazioni si posso vedere in **< UPDATE SET >** (quello che viene fatto dal programma)

STATO 0
UPDATE SET -0
STATO 1
UPDATE SET -1
.
.
FINAL STATE (e run terminate)

FINAL STATE: sono tutte la variabili che sono state usate nel MAIN

➔ Se le inizializzi e basta NON VERRANNO VISUALIZZATE.

RUN TERMINATE significa che il programma ha finito di girare e che quindi se inserisci qualcosa non succede niente(anche se buggato perché ti dice “EXPECTED a letter “)

MACRO RULE: non bisogna specificare nella funzione il ritorno ma nella **SIGNATURE** attraverso: static F: **Prod** (x,y) -> c