CIS 525 Parallel and Distributed Software Development

CONSTRUCTION OF O-GRAPHS

1. General idea:

a> CPN simulator = one marking at a time
b> OG tool = many markings at a time
c> Marking $\hat{=}$ set of pointers

2. Predefined functions of O-graphs

i> SearchModes (6 arguments)

ii> SearchArcs (6 arguments)

iii> OutArcs $\hat{=}$ a list of all the output arcs

iv> NodeProcessed $\hat{=}$ was node processed or not

v> Search Components $\hat{=}$ Searches SCC (6 arguments)
Used for automatic verification of the dynamic properties

vi> Standard queries:

→ "to verify whether transition is live"
→ "to verify whether an arbitrary set of bindings X ⊆ BE is live"
→ "user defined queries to 3 standard search functions"

3. Complexity of the O-graph

$$N_O(n) = N_O(n-1) + N_O(n-2)$$

Number of nodes for O-graphs

Where $N_O(2) = 3$
$N_O(3) = 4$

$$A_O(n) = 2 \times n \times F(n)$$

F(n) = Fibonacci sequence where F(2) = F(3) = 1

4. Experiments:

DataBase system with 8 DB managers

17.5 k nodes and 81.6 k arcs
60 min on Mc II fx and SUN 4
→ 60 MB memory
30 min on DEC station 5000/240

O-graph implemented in SML (Standard ML)

What is alternative to O-graphs?

- Lengthy / error prone testing and debugging

Options in O-graph construction!!!:

- whole or part of OG
- with or without time
- with or without code segments

OG tool is a separate program not related to CPN simulator

5. Hierarchical CP-nets

HL programming languages $\hat{=}$ subroutines, modules

Weakness of CP-nets $\hat{=}$ lack of compositionality

Hierarchical CP-nets = relate several CP-nets to each other in formal way that is well defined semantics (i.e. allows formal analysis)

Two hierarchy constructs:

- substitution transition
- fusion places

Relationships between CP-nets and hierarchical CP-nets