Control operators vs. Graph Logic Model

Igor Schagaev
London Metropolitan University, London, UK
i.schagaev@londonmet.ac.uk, also info@it-acs.co.uk

Abstract - *Simple paradigm to unite control operators for programming languages into one scheme using graph-logic representation of relations between agents (or elements of interaction) assuming independence of behavior for each element is presented. Shown that power of this structure exceed known models of description of behavior for concurrence and parallelism. Proposed model explicitly separates concurrency and parallelism and indicates further steps to automatic reprocessing programs for making them better tuned to modern architectures.*

Keywords: Computer languages, Control structures, Concurrent structures, Logic, Graph Theory.

1 Introduction

Every algorithmic language describes decision actions using logic statement of selection:

- "if" to choose one of two options
- "case of" to choose options from more than two
- "while" when our decision depends on conditions with uncertain time trigger or other independent parameter change

This is well supported by classification of relations introduced by E. Kant [1], whom I consider as a first theoretical programmer, opposing to a sentimental story of Ada, lady-lover of lord and part-time poet Byron. (Frankly, Nabokov's ADA makes much more sense to me).

E.Kant classified statements in terms of relationships and possible interactions between elements involved. Accordingly E. Kant an object might correspond, relate, and interact with others using the following relationships:

- *One-to-one*,
- *one-to-many*,
- *many-to-one*,
- *many-to-many*

And it seems to me nice and easy, provided we make decisions where to go, what to choose and our decisions are mutually exclusive. That is why, by the way, each processor instruction set has operator XOR and set-and-wait.

2 New Control Scheme

Unfortunately or fortunately our decisions are not always that simple as presented above: we can be friendly with different groups of people, make not mutually exclusive decisions, starts selective actions with various taboo: ""you can go your party but you do not drink and back before 11pm!" - remember? ;), etc., etc.

And this is all executes at the same time... Thus E.Kant diagram should be extended, one option of extension, called GLM is shown below on Figure 1. GLM stands for graph logic model to describe mutual dependency of various kind, was successfully applied in real world applications, including active conditional control systems, active safety monitoring systems, overpowers descriptive power of Markov and Bellman models and similar. Accordingly GLM, leaving one state, say "a" we might describe our leaving conditions using logic basic operators {AND, OR, XOR} attached to a leaving end of the links between "a" and neighbors.

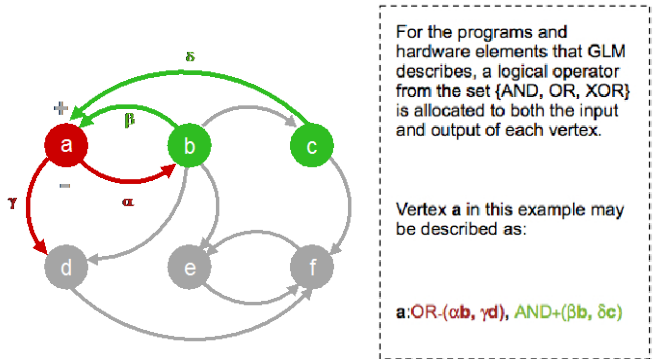

Figure 1 Graph Logic Model of interactions between nodes

Note that *at the same time* the various options are possible: selection of leaving conditions to several neighbors using {OR} on each out-coming link, or broadcasting to all neighbors through all out-coming links using {AND}, or picking just one and only neighbor using {XOR}.

Still, we think that we are the important ones and make impeccable decisions. This claim stands when we act and all others just follow…

But how about civil disobedience (for more on the subject I recommend to read Henry David Thoreau "On civil disobedience" or Gandhi passive resistance, when no matter what and which government (in first example US in the second UK) instructing how to obey and we do not follow and note – we act differently?

How to describe Vichy's collaborationism and De Gaulle resistance at the same time existing in France during WW2? How to describe Italian type strikes when people sit in office and do nothing? (Sound like EC...)

Did anybody spot - we are talking distributed computing now, as we have introduced various modes of reaction of opposite side of link and have to accept it's own will to act without our "instructions".

From now on the interactions between nodes with logic based decision rules applied to "own socket" for both sides of link are assumed!

All these examples of other nodes involvement in interaction force us to attribute the same link twice - at the leaving end and at the incoming end with different logic operators if necessary, Figure 2.

Figure 2 Logic operators for incoming and out-coming links

Using GLM we are able to describe mentioned above political phenomena and much wider and wilder conditions appeared in really of complex models without difficulties.

As another extreme example, when we assume that all logic operators within graph are operators are XOR we converge GLM into Markov model. In turn, adding weights (Greek letters on the graph of Figure 1) on links (cost, time other independent variables required) and assuming, again XOR as only operator allowed for out-coming and in-coming links we are able to describe Bellman optimization model using GLM notation.

Thus nodes and links between them with attributed logical operators attached to each leaving and coming ends form, in fact, new basis for control operators for next generation of programming languages.

3 Concurrency and Parallelizm

Almost everything that starts together, or at the same time or using the same data sooner or later will face a conflict of interests - parallel branches of program will require final aggregation of it into few numbers or functions; access to hardware, or informational, or time resources will be limited and conflict arises.

There are very few pure parallel program and systems - to name one known Sony PlayStation or any digital TV set - where incoming data flow splits and distributed in parallel to display visual elements.

For all the rest existing descriptive schemes of parallel program are not actually correct or useful. Use of GLM might help here:

What we start in parallel (leaving condition is AND for each link from a chosen node) might be completed in mutually exclusive mode (incoming condition XOR).

Using Figure 1 example traces **a-d-f** and **a-b-e-f** can be activated in parallel and eventually end up with conflict of interests, thus each of incoming links to node **f** should be attributed with XOR operators.

Tracing of branches of a program with attaching operators is becoming interesting area of research as we are able using GLM to separate really independent segments and allocate them properly on existing or next generation [2] hardware.

4 Conclusions

Graph logic model provides exceptional flexibility for expressing of control in various environments of interacting agents.

Attributed logical operators attached to each leaving and incoming ends of edge form new scheme of control operators for next generation of programming, when number of co-existing active agents will interact voluntarily.

5 Acknowledgements

Figure 1 was suggested and drawn by Simon Monkman, - my former student and good friend. Value your support, Simon.

6 References

[1] E. Kant "Critics of pure reason", Everyman ISBN 0-40-87357-X, 1993

[2] Schagaev et al. ERA – Evolving Reconfigurable Architecture. ERA Proc. of Conf: Software Engineering, Artificial Intelligence, Networking. and Parallel/Distributed Computing – SNPD 2010, PP 215-220.