Calculus inference graphs and analog circuit synthesis

Pierre Fouilhoux, S. Kedad-Sidhoum[1]

February 5, 2008

Abstract

In this article, we consider a calculus system as a set of variables linked by calculus operations. These operations can be algebric equations, differential equations, algorithmic functions, boolean equations... We interest in finding a particular set of variables, called parameters, such that the values of the resting variables can be infered by using the operations. This problem reduces to a combinatorial optimization problem, called the calculus inference subgraph problem, for which we propose both exact and approximative solving methods. We focus on applying these results to analog circuit synthesis in order to compute the component sizing variable values from the expected values of both input and output circuit parameters.

Keywords:

1 Introduction

Let $\mathcal{V} = \{v_1, ..., v_n\}$ be the set of numerical *variables* associated to a domain D_{v_i}, $i = 1, ..., n$, of possible numerical values. Let $\mathcal{O} = \{o_1, ..., o_m\}$ be a set of *calculus operations* (for instance algebric equations, differential equations, algorithmic functions, boolean equationns ...) associated with a subset $L_j \subset \mathcal{V}$, $|L_j| \geq 2$, of variables. The couple $\mathcal{S} = (\mathcal{V}, \mathcal{O})$ is then called a *calculus system.* We say that a numerical vector $s \in I\!R^n$ is a *solution* of $\mathcal{S}$, if the value $s(v_i) \in D_{v_i}$, $i = 1, ..., n$, can be associated to variable v_i such that each of the operations o_j, $j = 1, ..., m$, is numerically *satisfied.* Solving a calculus system $\mathcal{S}$ consists in determining a subset $\mathcal{P} \subset \mathcal{V}$, called the *parameter* set such that, for a set of numerical values $s(v) \in D_v$, $v \in \mathcal{P}$, one can *infer* (ie, easily compute) the values $s(v) \in D_v$, $v \in \mathcal{V} \setminus \mathcal{P}$ such that s is a solution of $\mathcal{S}$.

EXAMPLE linear system (with m¡n and Gauss reformulation), equadiff coming from mecanics,... There exists framework where $\mathcal{P}$ is an emptyset: ex linear system with n variables and m linearly independent constraints.

In this article, we focus on calculus systems that can be directly solved by using properly the operations without any transformation on the system. In that purpose, we will show how to compute an *inference sequence* as a sequence $(o_{\delta(1)}, ..., o_{\delta_m})$ of operations such that the values of variables of $\mathcal{V} \setminus \mathcal{P}$ can be computed from values of $\mathcal{P}$ by using the operations along that sequence. On can notice that we of course will not deals with numerical values but on a way to calculate them. In that purpose, we only interest in calculus system such that for every operation o_j, $j \in \{1, ..., m\}$, there exists at least one subset I_j, $|I_j| \geq 1$ of L_j such that, if a numerical value is given to each variable of I_j, then we know how to compute some values of $L_j \setminus I_j$ from values of I_j so that operation o_j is satisfied. By this property we say that an operation can be used as an *operator* that infers (deduces by computation) the values of $L_j \setminus I_j$ from values of I_j. This assumption is not too restrictive as most of calculus systems follow these description. In fact, if it is not the case, it is hard to solve them with exact methods and simulation process are then

[1]Laboratoire LIP6, Université Pierre et Marie Curie, 4 place Jussieu, 75005 Paris, France. e-mail:Pierre.Fouilhoux@lip6.fr, Safia.Kedad-Sidhoum@lip6.fr

necessary.

To give an illustration, we can see a calculus system as hypergraph where variables are the nodes and operations are the (hyper)-edges as on Figure...We then can easily see how the calculus system is organized. An inference sequence will then be a sub-hypergraph...

We then will define the inference sequence as a sequence of operators. If we can give such a sequence, then, the vector s infered from a set of values $s(v), v \in \mathcal{P}$, satisfies all the operations. Then if $s(v_i)$ belongs to domain D_{v_i} for every $i \in \{1, ...n\}$, then s a solution of S. Unfortunately, it can be sometimes to use all the operations as operators in an inference sequence. In that case, it should not be sure that the infered vector s is a solution. We then will interest in finding an inference sequence that contains a maximum number of operations. We will show that this problem can be reduced to the *inference calculus subgraph problem*.

Biblio inference graph: logic / le truc du schrijver / les listes topologiques / les graphes acyclic /...

In addition with the maximum number of operations taken in the inference sequence, we will consider another criterion. In fact, in analog circuit synthesis, some of the variable values can be fixed by the circuit designer. Since, the automatic process have to fixed a huge number of variables, we want to ask the designer to fix the least possible number of variables. So among the inference sequence which contains the maximum number of operations, we will search those whose possess the minimum number of parameters.

Biblio????

Dans cet article, nous tudions le problme de dtermination d'une liste d'infrence de calculs comme un problme d'optimisation combinatoire. Nous proposons une formulation PLNE de ce problme dans le cas gnral. Nous proposons galement des heuristiques pour rsoudre ce problme dans le cas de graphes d'infrence de grande dimension. Nous proposons galement des cas particuliers de graphe o le problme est polynomial.

Notations

2 Modelisation

In this section, we will show how the problem of finding an inference sequence reduces to the so-called calculus inference subgraph problem.

For a given operation o_j, we associate l_l *operators* $o_j^k = (I_i^k, O_i^k)$, $k = 0, ..., l_j$, where $I_j^k \cup O_j^k$ is a partition of L_j. Each sense o_j^k corresponds to a known way to apply the operation o_i to compute the values of of O_j^k variables from the values of the I_j^k variables. According to the type and difficulty of an operation, it should exist one, several or all the possible opetators. In practice, of course, $l_j \leq 1$. We can remark that a sense when $O_j^k \neq \emptyset$ or I_k^k is impossible, and the cases when $|I_k^k| < |O_j^k|$ seems to be rare.

EXAMPLE. Remarque sur les operations coincidence.

EXAMPLE, equation où on peut fixer 2 variables pour en avoir une 3eme, mais a ne marche pas en fixant seulement 1 variable. EXAMPLE, une equadiff super dure qu'on ne sait pas resoudre correctement dans tous les sens (venant de CAO). CONTRE-EXEMPLE Newton-Raptson ie mauvaise lecture des operations.

Le graphe $H = (V, A \cup E)$ d'infrence de calcul se dfinit partir de $\mathcal{P}$ et de $\mathcal{O}$ de la faon suivante. Un sommet de V correspond uniquement soit un paramtre de $\mathcal{P}$, soit au sens d'un oprateur de $\mathcal{O}$. Pour un sens $o^k = (E^k, S^k)$ d'un oprateur o, il existe un arc (p, o^k) (resp. (o^k, p)) dans A si p est un paramtre de E^k (resp. de S^k). Il existe une arête pp' dans E si les paramtres p et p' coïncident.

Le problme dcrit dans l'introduction revient donc dduire de H une arborescence contenant tous les sommets-paramtres, exactement un sommet-sens par oprateur et qui contienne un nombre maximum d'arêtes. De plus, on recherche une arborescence contenant un nombre minimum de sommets sans prdcesseur.

La figure illustre les difficults d'extraire une chaîne d'infrence cohrente. On remarque que les paramtres p_2 et p_4 sont relis par une arête et doivent a priori coincider, c'est--dire être gaux, ce qui est impossible en permanence si la liste d'infrence dduite de H contient le sens o_1^1. En effet, dans ce cas, il existe une chaîne calculant p_4 partir du paramtre d'entre p_5 et une chaîne calculant p_2 partir du paramtre p_1, qui devient alors paramtre d'entre. Si en revanche on choisit le sens o_1^2, cette incohrence n'existe plus et de plus, p_1 n'est pas un paramtre d'entre. En revanche, ce choix induit que le sens o_4^2 est choisi et que p_8 devient paramtre d'entre. On peut galement remarquer un deuxime type d'incohrence. Les sommets $(o_3^1, p_7, o_5^1, p_6, o_2^1, p_4, p_2)$ forment un cycle compos d'arcs et d'arêtes. Si on choisit alors le sens o_3^2, on a finalement dtermin une liste d'infrence issue du graphe o p_5, p_6, p_8 et p_9 sont des paramtres d'entres.

3 Calculus inference subgraph

4 Les conflits: thm pour passer des conflits aux calculus inference subgraph

5 Polynomial cases

6 PLNE

7 Heuristics

8 Application la synthse de circuits analogiques

Les graphes d'infrence de calcul se rencontrent dans le processus de synthse des circuits lectroniques analogiques. Un tel circuit est dfini par un ensemble de composants en silicium relis les uns aux autres. Chaque composant est dtermine par un ensemble de paramtres de conception (largeur, longueur,...) dterminer en fonction de son utilisation dans le circuit. Pour chaque composant, il existe un ou plusieurs oprateurs liants les paramtres de conception et des paramtres locaux du circuit: tension un point donn du circuit, intensit d'un fil, etc. Le circuit induit alors un large graphe d'infrence entre tous ces paramtres par l'intermdiaire de ces oprateurs. Les paramtres situs aux extrmits du circuit sont ainsi les paramtres du circuit tout entier: il s'agit bien entendu des signaux (tension, intensit,...) d'entre et de sortie du circuit, mais galement des paramtres mesurant les performances du circuit. À partir des valeurs attendues pour les paramtres de sortie et de performance, en fonction de paramtres d'entre, les concepteurs de circuits analogiques dsirent calculer toutes les valeurs des paramtres du circuit.

La fixation de tous les paramtres de conception de chacune des composantes est difficile dduire du graphe d'infrence des calculs correspondant un circuit. Les logiciels de conception utilisent un moteur de simulation qui ajuste les paramtres progressivement. L'environnement logiciel CAIRO+ [1, 2] propose de dduire une liste d'infrence des calculs cohrente. Pour cela, CAIRO+ demande au concepteur du circuit d'ajouter des connaissances, comme par exemple la fixation de la valeur de la tension attendue en un point prcis du circuit. Le problme pos par CAIRO+ se ramne ainsi l'extraction de la meilleure liste d'infrence possible (avec le moins possible d'incohrences) et qui demande un nombre minimal d'ajout de connaissances pour le concepteur.

Remerciements: Les auteurs remercient Marie-Minerve Louërat, Ramy Iskander et Dimitri Galayko pour leurs prcieux conseils.

References

[1] Iskander R., Louerat M.-M., Kaiser A.: Hierarchical Graph-Based Sizing for Analog Cells Through Reference Transistors. Research in Microelectronics and Electronics, pp 321- 324 (2006).

[2] Iskander, R., Kaiser, A., Louerat, M.-M.: Automatic biasing point extraction and design plan generation for analog IPs. Circuits and Systems. 48th Midwest Symposium on Volume, pp 907-910 Vol. 1 (2005).