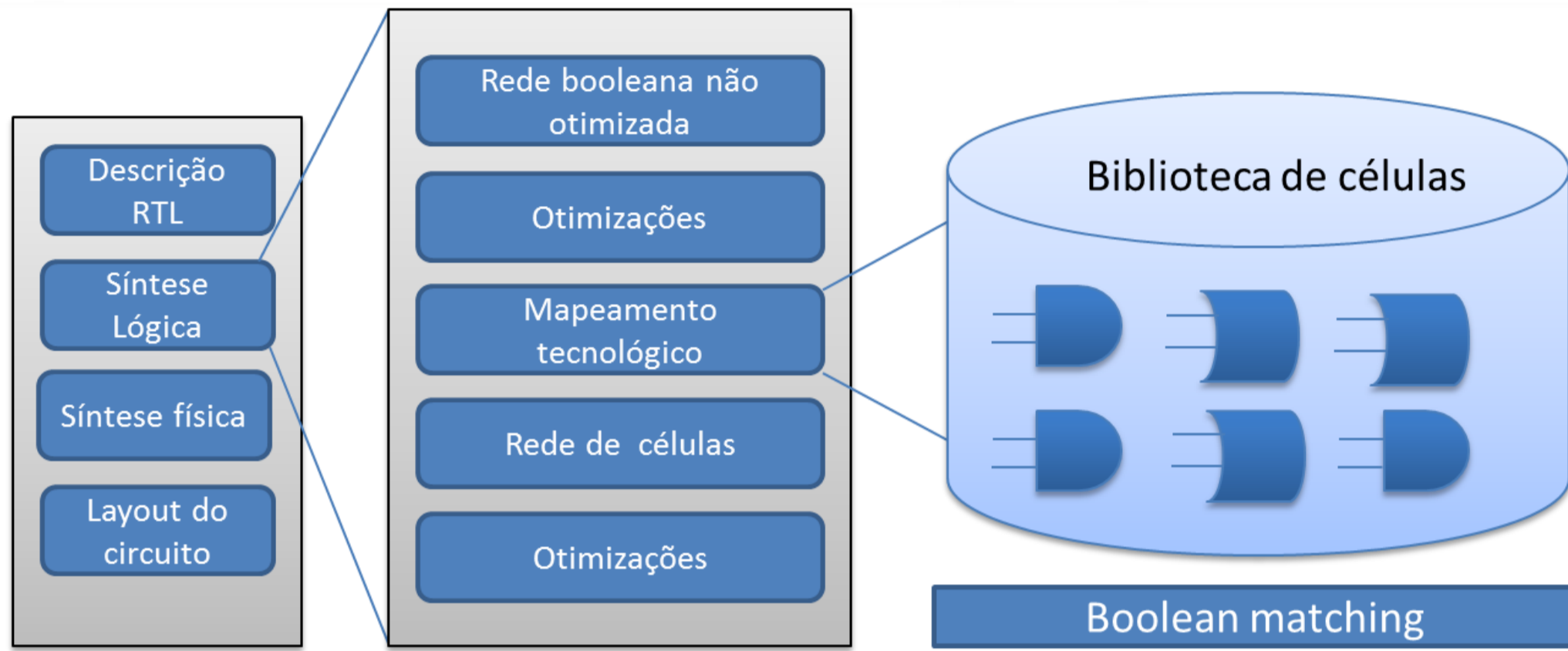


Anderson Santos da Silva, Vinicius Callegaro, Renato P. Ribas, André I. Reis
 Instituto de Informática / PPGC - UFRGS, Porto Alegre, Brasil
 {assilva, vcallegaro, rpribas, andreis}@inf.ufrgs.br

INTRODUÇÃO

O fluxo 'standard cell' desempenha um papel importante no projeto de circuitos digitais. Este fluxo é dividido em várias etapas. Uma delas é o processo de mapeamento tecnológico, que recebe um 'netlist' e cobre candidatos junto a células de uma biblioteca. Como há diversos cortes em um 'netlist', muitos deles devem ser considerados. Claramente, tal tarefa precisa ser tão rápida quanto possível.



Fluxo 'standard cell' para projeto de ASIC.

BOOLEAN MATCHING:

- a) Usado na etapa de mapeamento tecnológico
- b) NP-completo para funções em geral

Equivalência-P ('matching'): Tipo de equivalência Booleana onde duas expressões lógicas são equivalentes-P se uma permutação (P) sobre suas variáveis de entrada puder transformar uma expressão na outra.

A	B	C	F ₁	A	B	C	F ₂
0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	1	1	0	0	1
1	0	1	0	1	0	1	0
1	1	0	1	1	1	0	0
1	1	1	0	1	1	1	0

$$F_1(A,B,C) \neq F_2(A,B,C)$$

$$F_1(B,C,A) = F_2(A,B,C)$$

Exemplo de equivalência-P.

Abordagens para funções em geral:

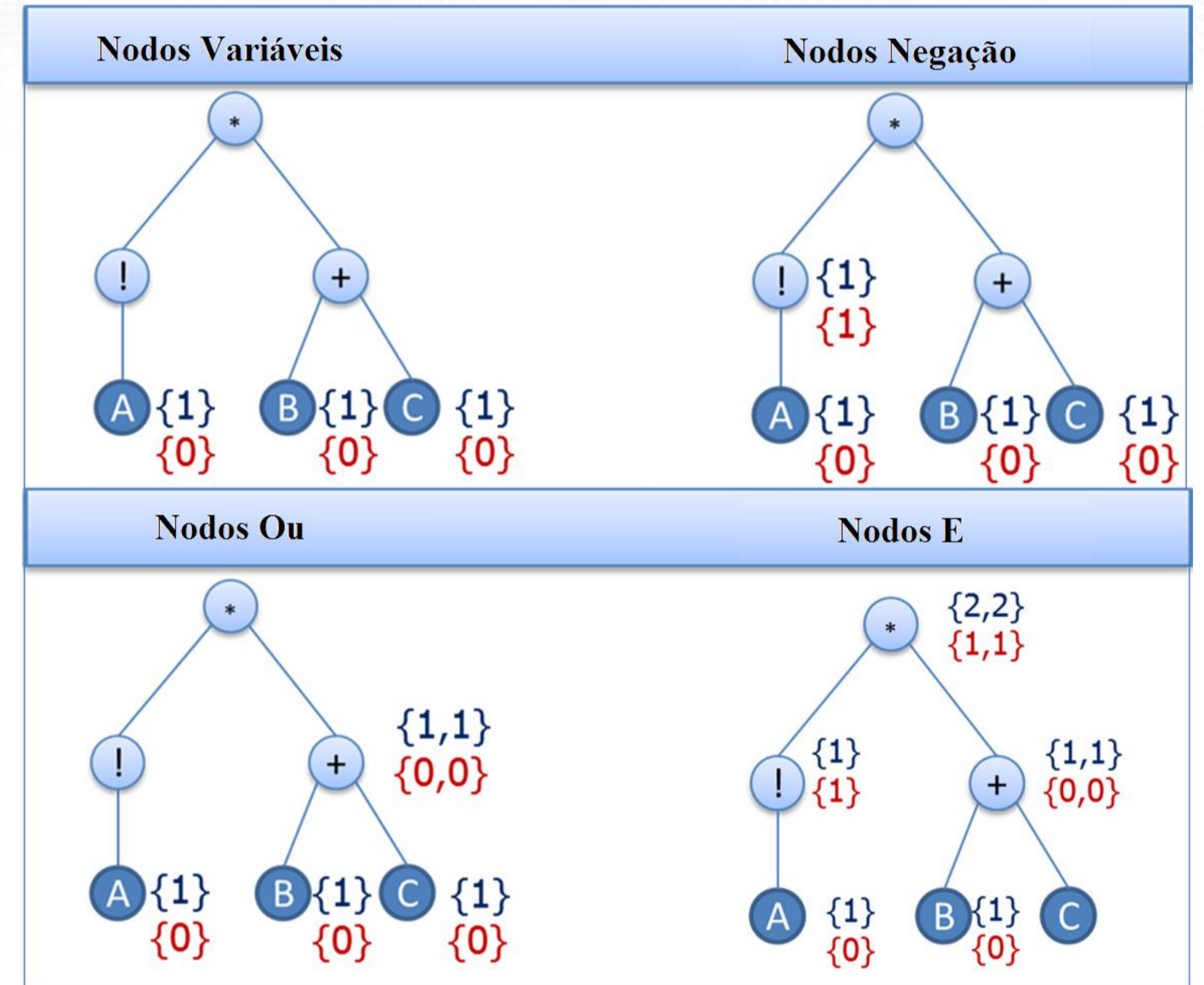
- a) Sasao & Butler [1]
 - 7 variáveis é o limite máximo
 - complexidade de memória: $O(2^n \times n!)$
 - complexidade de tempo : $O(2^n \times n!)$
- b) Heinsberger & Kolla [2]
 - complexidade de memória: $O(2^n)$
 - complexidade de tempo: $O(2^n \times n!)$

REFERÊNCIAS

- [1] T. Sasao and J. T Butler, "Progress in Applications of Boolean Functions," *Synthesis Lectures on Digital Circuits and Systems*, vol. 4, no. 1, 2009, pp. 1-153.
- [2] U. Hinsberger and R. Kolla, "Boolean matching for large libraries," *In Proc. Design Automation Conference (DAC)*, Jun. 1998, pp. 206-211.

ESTRATÉGIA

Uma função Booleana é considerada 'read-once'(RO) se esta pode ser representada por uma forma fatorada onde cada variável aparece somente uma vez. Funções RO são abundantes em circuitos. Logo, a idéia básica é acelerar o processo de equivalência Booleana usando as propriedades destas funções.

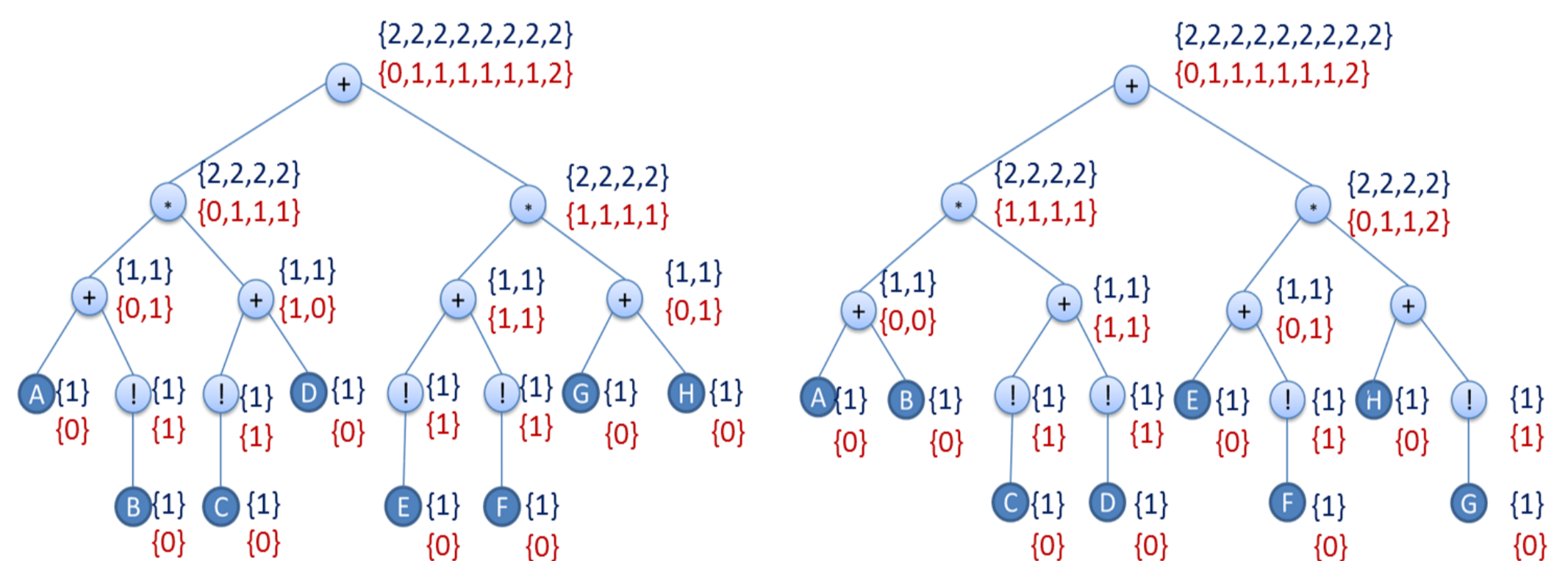


Regras de normalização

ALGORITMO

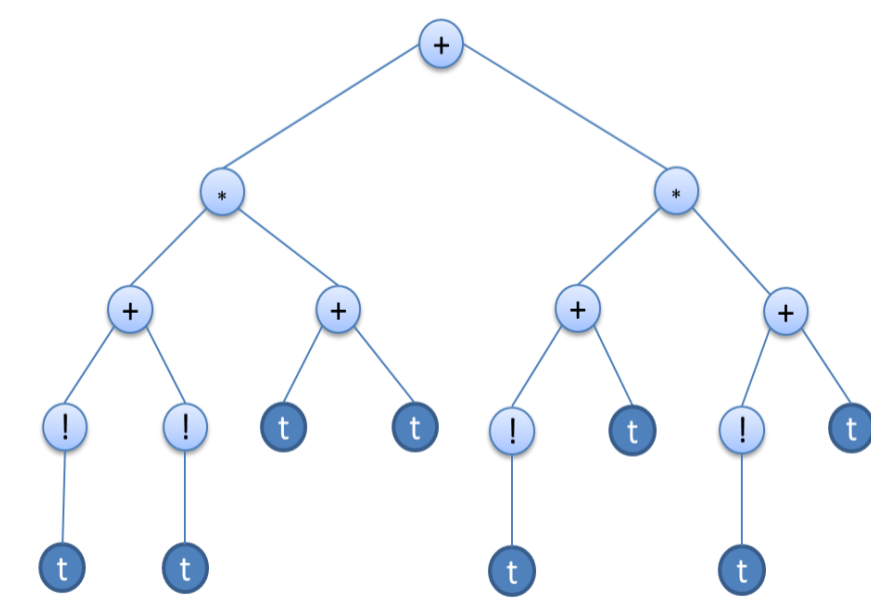
Usa idéia de normalizar a representação em árvore de uma função RO usando sua estrutura canônica..

- Codifica a árvore:



Codificação da árvore

- Passa sub-árvores mais densas para o lado mais à esquerda:



Árvore normalizada

- O percurso em profundidade à esquerda constrói uma assinatura para a árvore:

$$(((t + !t) * (t + t)) + ((!t + t) * (!t + t)))$$

Se duas assinaturas forem iguais, as árvores serão equivalentes.

CONCLUSÃO

O método proposto recebe duas expressões RO e normaliza elas baseado em sua representação em árvore. Um algoritmo com complexidade na ordem de $O(k^2 \times s)$ foi desenvolvido.

AGRADECIMENTOS

Pesquisa parcialmente financiada pela Nangate Inc. em um acordo de pesquisa Nangate/UFRGS, pelas agências de financiamento Brasileiras CAPES e CNPq, pela FAPERGS ("grant" 11/2053-9 - Pronem) e pela Comunidade Europeia "Seventh Framework Programme" ("grant" 248539 - Synaptic).