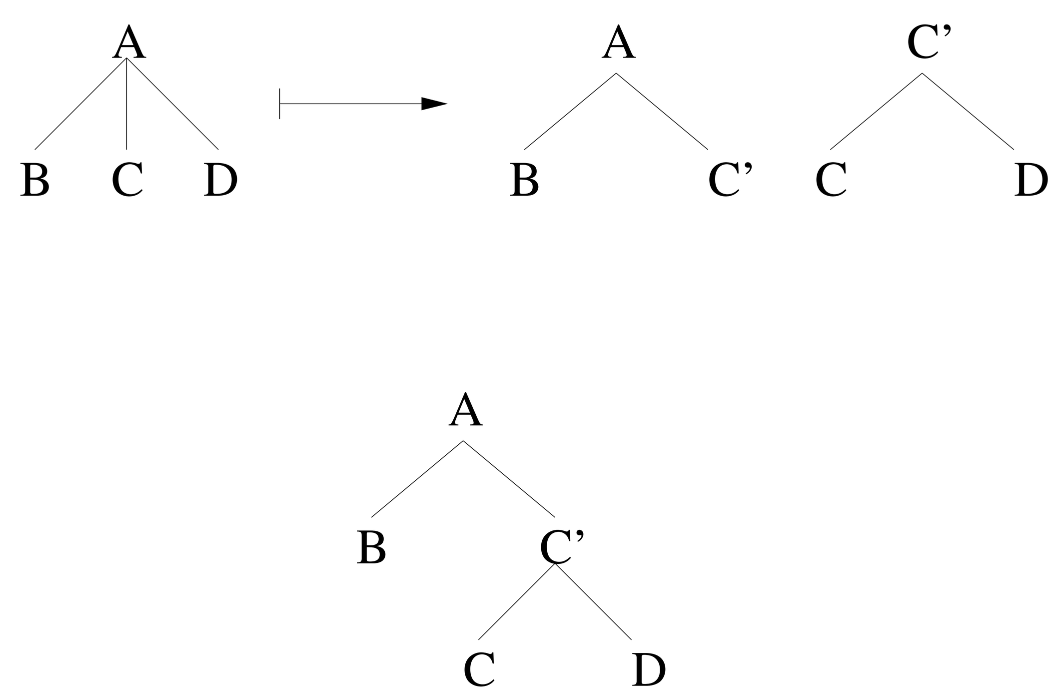
# A CNF Style Transformation For Multidimensional Trees

David Brown, Ian Kelly, Colin Kern, Alex Lemann, Greg Sandstrom
Earlham College, Department of Computer Science

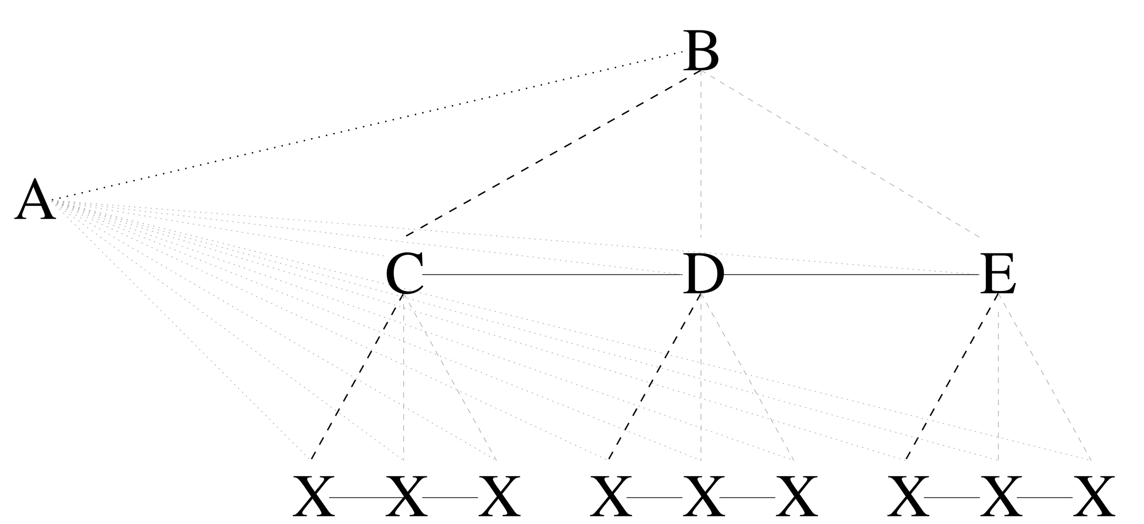
## Abstract

We introduce a transformation for factoring arbitrary branching multidimensional local trees into sets of strictly 2-branching local trees, resembling the conversion to Chomsky Normal Form for 2-dimensional trees. We explore the size of local multidimensional trees with constant branching factor and show that this is linear in the number of dimensions for complete 2-branching local trees and hyper-exponential in the number of dimensions for complete 3-branching local trees. Hence our transformation entails a blow-up in grammar size by a factor that is hyper-exponential in the dimension. We show that our transformation is optimal with respect to this asymptotic bound.

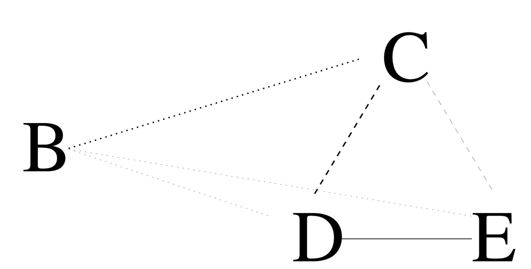
## Chomsky Normal Form (CNF) Transformation



## A Full 3–Branching 3–Dimensional Tree



## A Full 2–Branching 3–Dimensional Tree

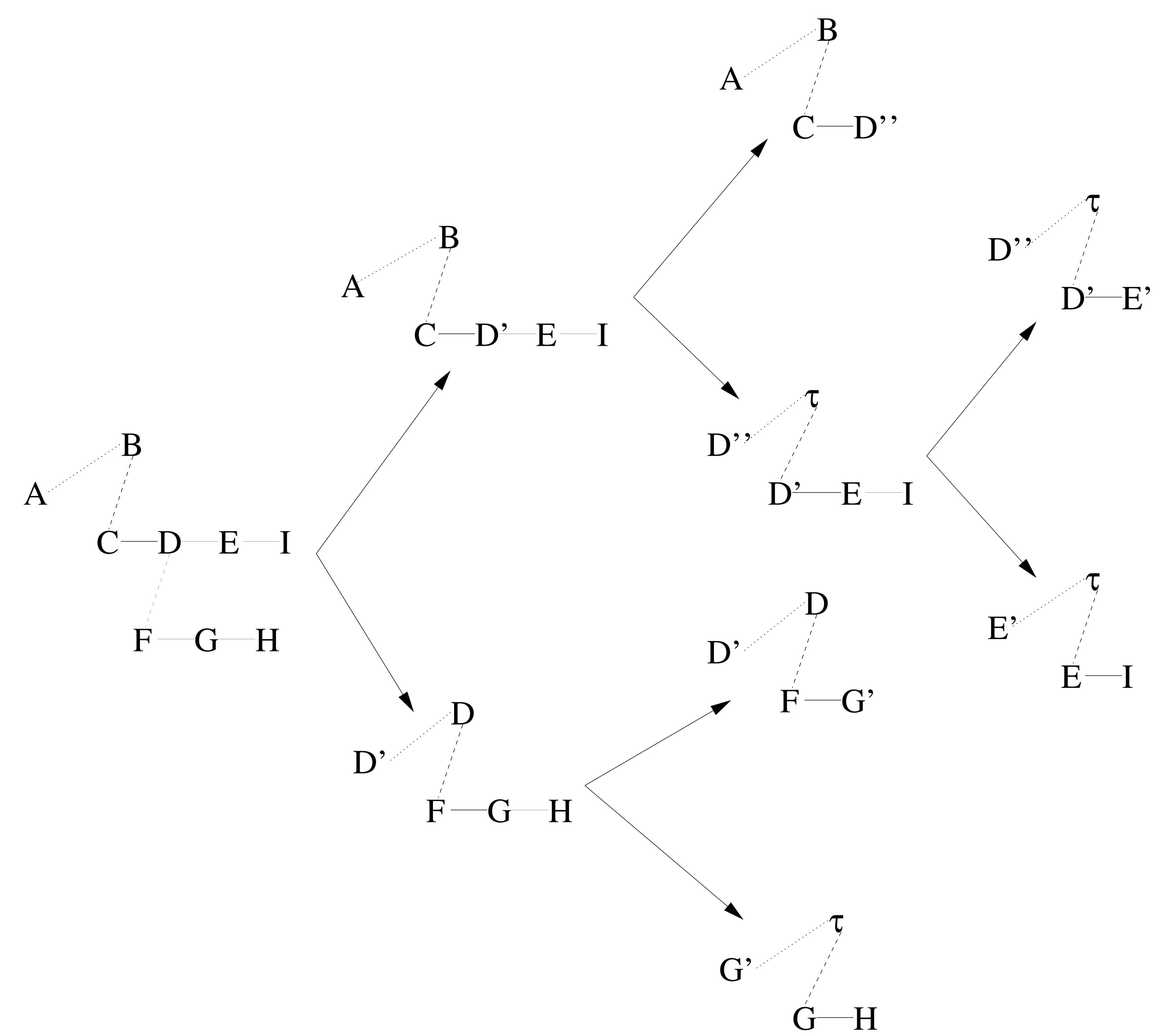


## Outline of the Transformation Algorithm

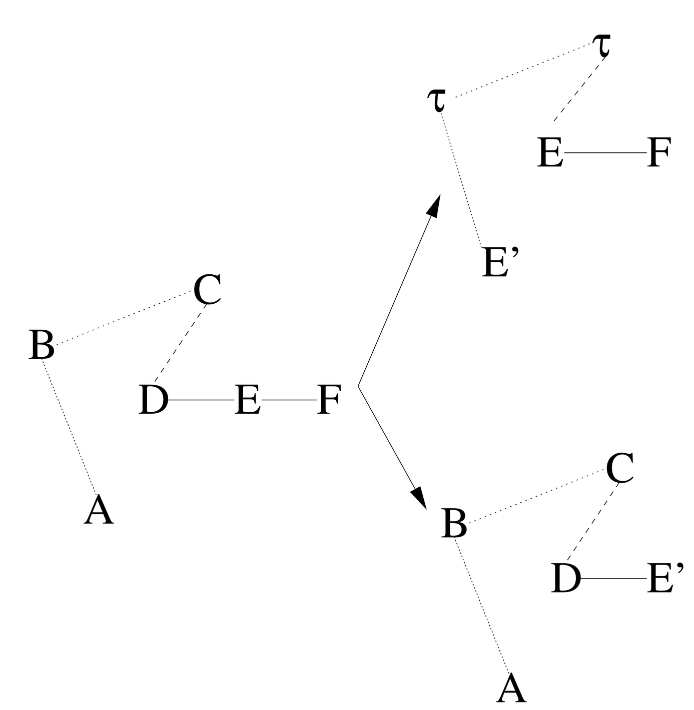
The transformation algorithm:

- Traverse the tree in a depth-first method.
- For every node, check each link for a successor that breaks the 2-branching definition.
- For every such successor, split the tree into two trees:
  - A tree with the subtree rooted at the successor removed, with the current node renamed to a unique label.
  - The subtree rooted at the successor with the current node at the root.
- Use $\tau$ nodes to fill out the nodes of a new tree if the dimension is less than the dimension of the original tree.
- Repeat on each factored tree until no more factors are created.

## Example Factoring of a 3–Dimensional Tree

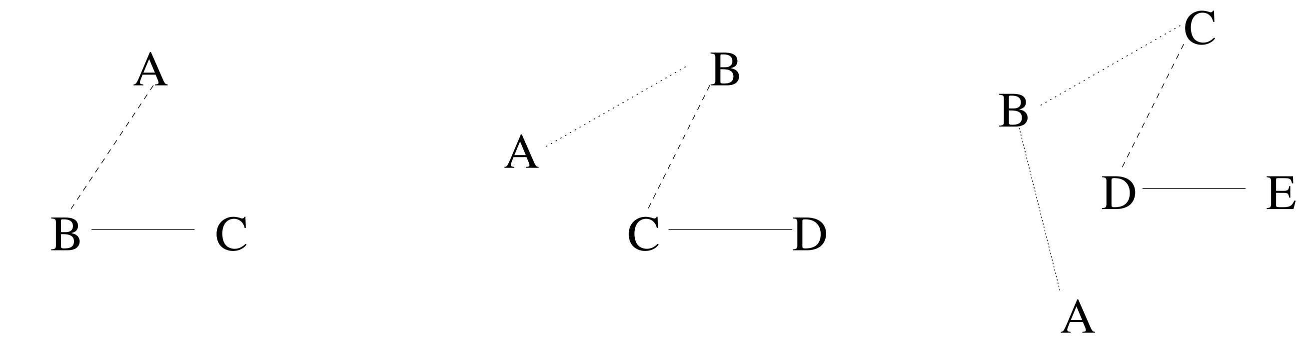


## The Use of 'tau' Nodes in Factoring a 4–Dimensional Tree



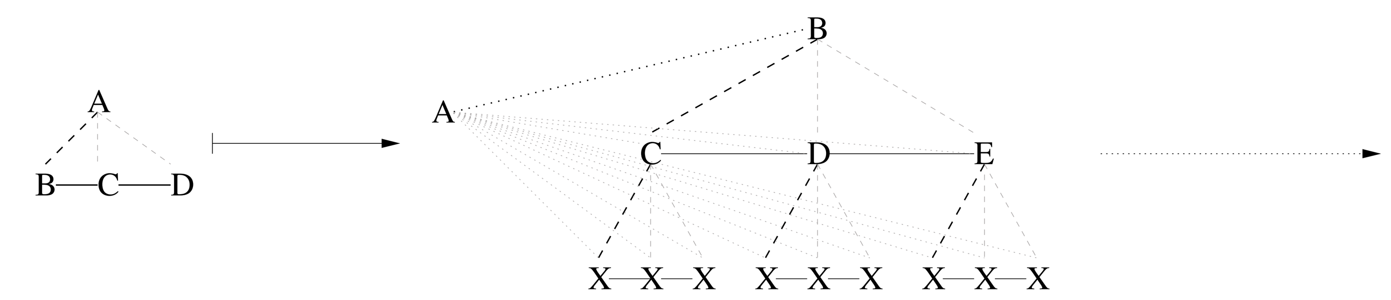
## Growth in Tree Size of 2–Branching Trees As Dimension Increases

2 dimensions 3 dimensions 4 dimensions



\# of nodes = # of dimensions + 1

## Growth in Tree Size of 3–Branching Trees As Dimension Increases



## Growth of 3–Branching Trees as a Recursion

$$N_3(0) = 1$$
$$N_3(d) = N_3(d-1)^2 - N_3(d-1) + 2$$

We can solve this recursion:

$$N_3(d) = \Omega(k^{(2^{(d-1)})})$$

The growth is hyper-exponential in the dimension

## Theorem For the Growth of Grammars

**Theorem 1** *For a full $d$-dimensional, $n$-branching local tree, the number of local trees in the factored form required is equal to the number of 1-dimensional illegal links.*

Before we can prove this, we need to establish some lemmas.

**Lemma 1** *In a full $n$-branching tree, when an $i$-dimensional link is fixed using embed, a $j$-dimensional link is also fixed for all $0 < j < i$.*

**Lemma 2** *Let $l(i)$ be the number of $i$-dimensional links that are not fixed by the side-effect of fixing higher dimensional links as in Lemma 1 and $t(i)$ be the total number of $i$-dimensional illegal links, then $l(x) = t(x) - t(x+1)$.*

## Future Work

We are working on the converse of the embed operation, which we call "excavate". "Excavate" will take a tree constructed with 2-branching trees and produce the equivalent tree constructed with the arbitrary branching trees of the original grammar.

While we are confident in the correctness of *embed*, we have not provided a formal proof. When we have finished developing *excavate*, we expect to simultaneously prove the correctness of both algorithms by proving

$$T(G) = excavate(T(embed(G)))$$

where $T(G)$ denotes the set of trees licensed by the grammar $G$.

This research made possible in part by the Ford–Knight Research Grant, the Matthews Summer Research Fund, and the Earlham CS Collaborative Research Fund.