

The Christmas Tree Dilemma

Sarah Berkemer

Bioinformatics Leipzig

Winterseminar 2016, Bled

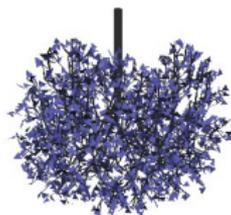
Introduction



This is a Tree..

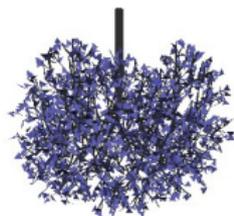
Definitions

This is a Tree..

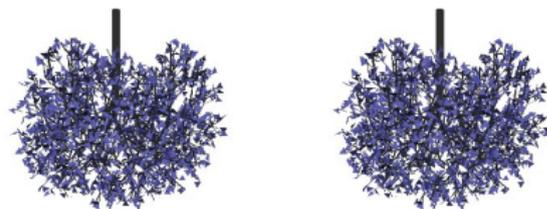


Definitions

This is a Tree..

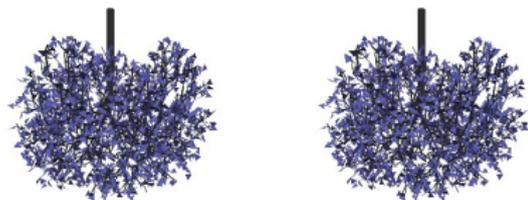


..and this is a Forest.



Definitions

This is a Forest..

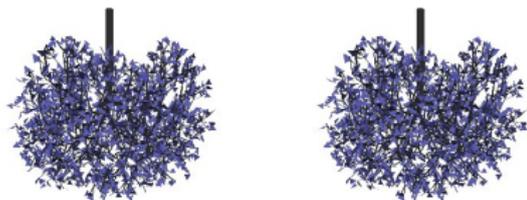


..and this is a Forest, too.



Definitions

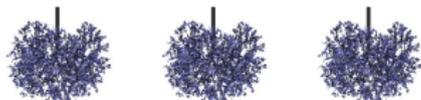
This is a Forest..



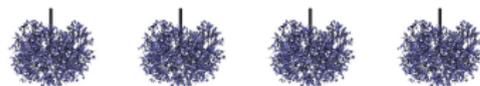
..and this is a Forest, too.



..and this..



..and this, too.



Introduction II

This are christmas trees...



Definitions

This is a Christmas tree..



Definitions

This is a Christmas tree..



..and this, too.



Definitions

This is a Christmas tree..



..and this, too.



This is a Christmas forest..

© 2012 Sarah Berkemer

Definitions

This is a Christmas tree..



..and this, too.



This is a Christmas forest..



..and this, too.



Introduction III

These are Christmas ornaments..



Introduction III

These are Christmas ornaments..



..and this, too.



Introduction III

These are Christmas ornaments..



..and this, too.



And this..



Introduction III

These are Christmas ornaments..



..and this, too.



And this..



..and this, too.



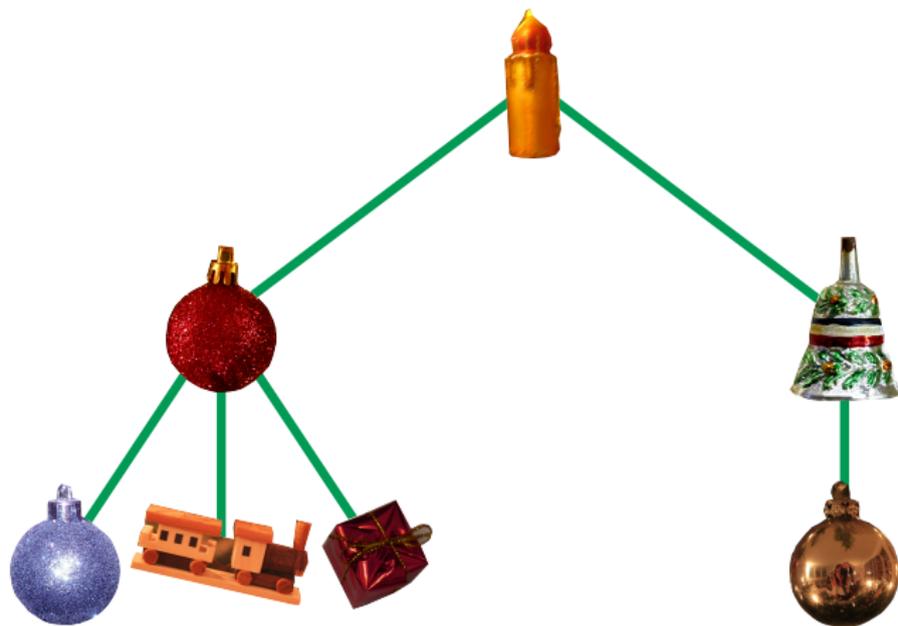
Basics

Assume that

- christmas tree = tree
- christmas forest = forest
- christmas ornament = label



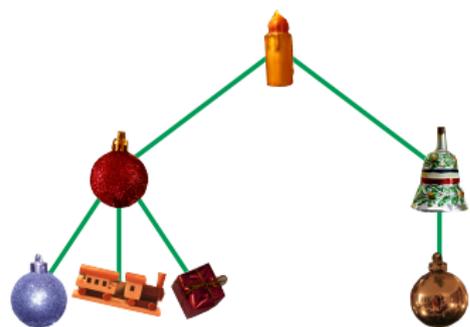
Our Christmas Tree



The Neighbours' Tree



Higher, better, nicer, stronger..



?
⇒

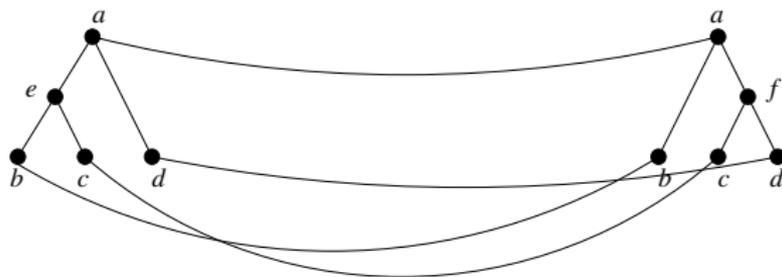


Tree Editing

Definition

A mapping between two ordered trees (forests) T_1 and T_2 is a binary relation on pairs of vertices (x, y) and (x', y') with $x, x' \in V(T_1), y, y' \in V(T_2)$ such that the following conditions hold:

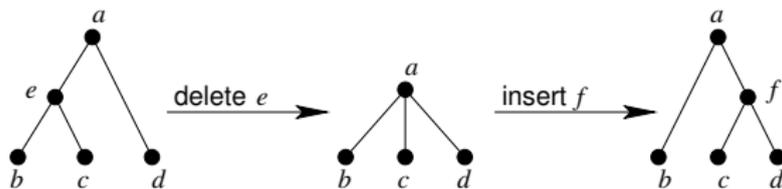
- one-to-one condition: $x = x' \Leftrightarrow y = y'$
- ancestor condition: x ancestor of $x' \Leftrightarrow y$ ancestor of y'
- sibling condition: x left sibling of $x' \Leftrightarrow y$ left sibling of y'



Tree Editing

Operations

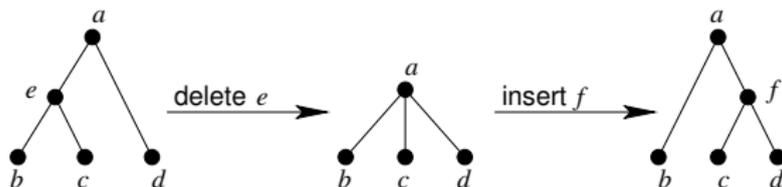
- relabeling
- deletion
- insertion



Tree Editing

Operations

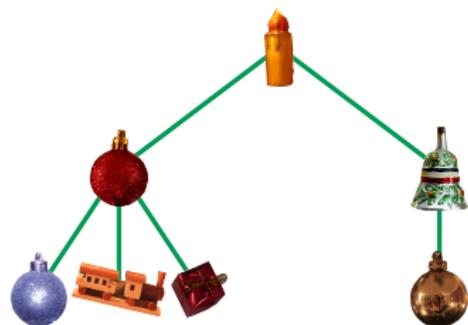
- relabeling
- deletion
- insertion



Aim: Minimize number of operations!

⇒ can be done with a DP algorithm.

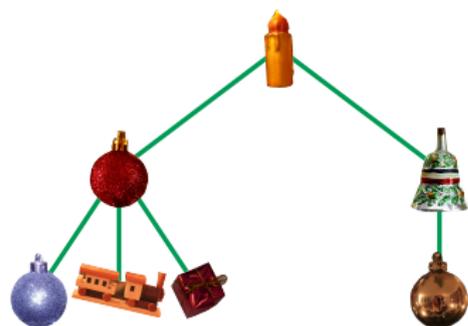
Tree Editing



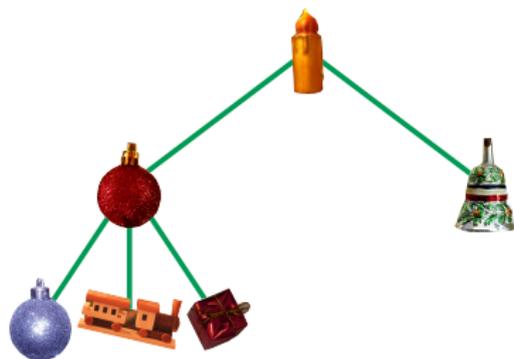
⇒
Insert 
Insert 
Insert 
Insert 
Delete 
⇓



Tree Editing



⇒
Insert 
Insert 
Insert 
Insert 
Delete 
⇓



Tree Alignment

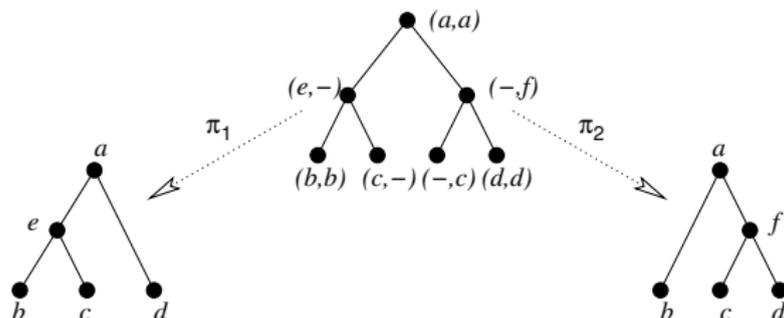
Definition

A mapping of labels on forests F_1 and F_2 based on $(\mathcal{A} \cup \{-\}) \times (\mathcal{A} \cup \{-\})$ and restrictions $\pi_1(G)$ and $\pi_2(G)$ by considering either the first or the second coordinate, with G being the resulting alignment forest.

Tree Alignment

Definition

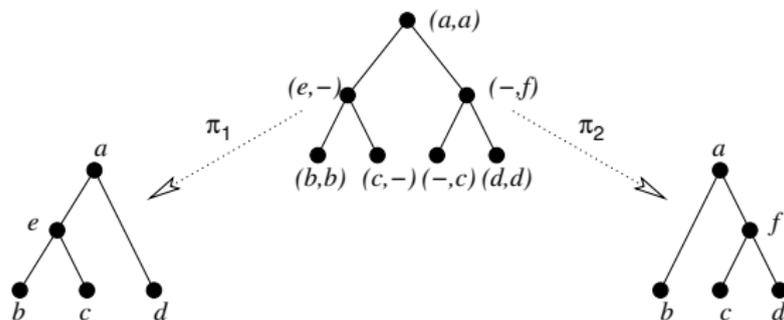
A mapping of labels on forests F_1 and F_2 based on $(\mathcal{A} \cup \{-\}) \times (\mathcal{A} \cup \{-\})$ and restrictions $\pi_1(G)$ and $\pi_2(G)$ by considering either the first or the second coordinate, with G being the resulting alignment forest.



Tree Alignment

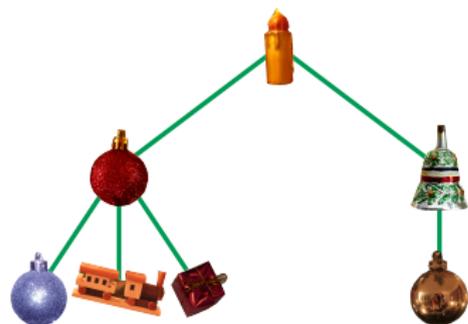
Definition

A mapping of labels on forests F_1 and F_2 based on $(\mathcal{A} \cup \{-\}) \times (\mathcal{A} \cup \{-\})$ and restrictions $\pi_1(G)$ and $\pi_2(G)$ by considering either the first or the second coordinate, with G being the resulting alignment forest.

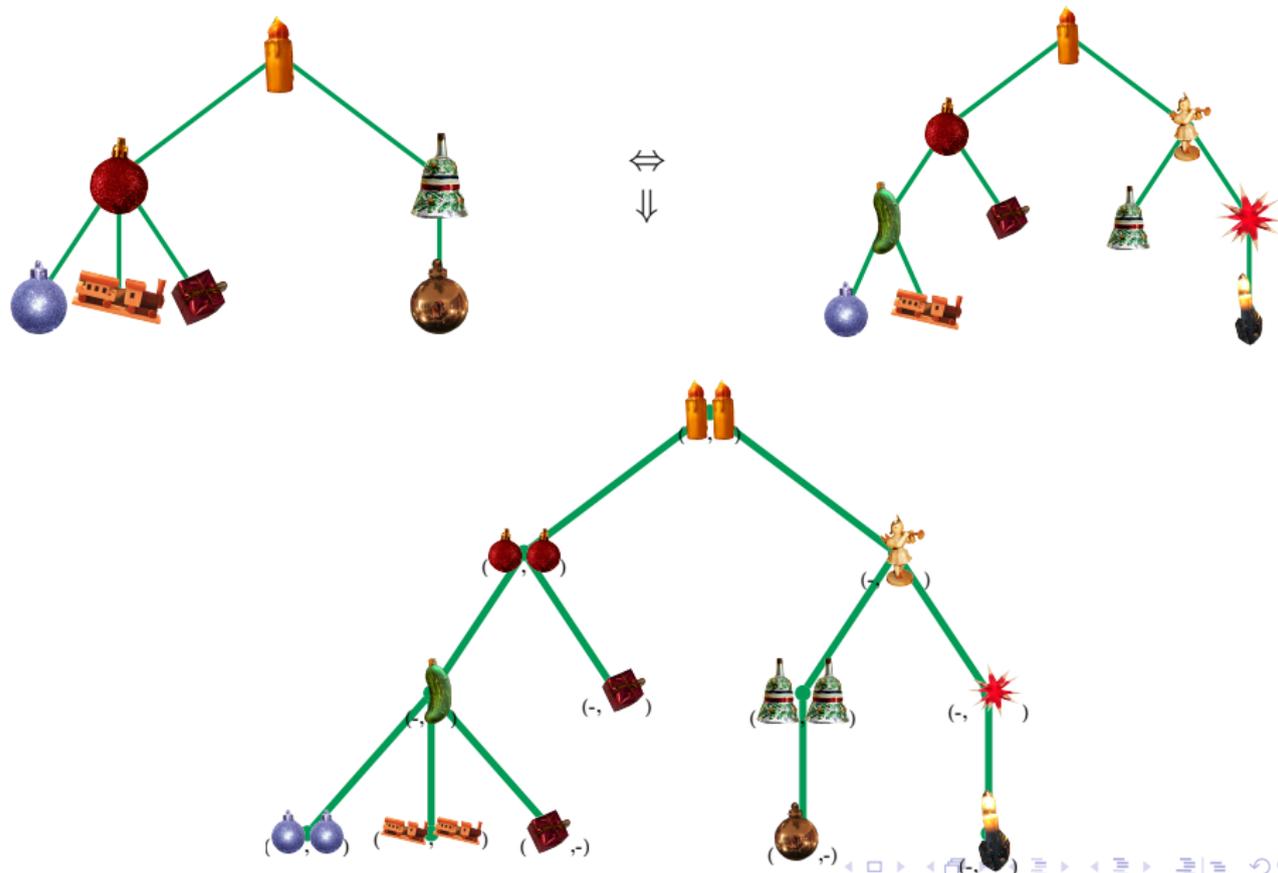


The cost of the alignment is the sum of the cost of label pairs.
 \Rightarrow can be done with a DP algorithm.

Tree Alignment



Tree Alignment



Grammars

Definition

Context-free grammars with production rules of the form $V \rightarrow \alpha$, where V is a non-terminal and α is a string of terminals and/or non-terminals.

Grammars

Definition

Context-free grammars with production rules of the form $V \rightarrow \alpha$, where V is a non-terminal and α is a string of terminals and/or non-terminals.

The Formula

grammar + scoring algebra + index structure = DP over arbitrary data structures

Grammars

Definition

Context-free grammars with production rules of the form $V \rightarrow \alpha$, where V is a non-terminal and α is a string of terminals and/or non-terminals.

The Formula

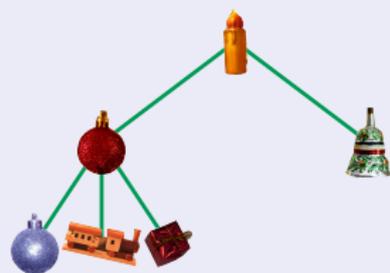
grammar + scoring algebra + index structure = DP over arbitrary data structures

The Task

- 1 find a grammar describing your problem
- 2 find a scoring algebra
- 3 find an index structure and iteration rules for your data structure (if not done yet)

Grammars

Traversing Forests



T →



F →



n



F



T



F

Tree Editing

$$\left(\begin{smallmatrix} S \\ S \end{smallmatrix}\right) \rightarrow \left(\begin{smallmatrix} F \\ F \end{smallmatrix}\right) \quad [\textit{start}]$$

$$\left(\begin{smallmatrix} F \\ F \end{smallmatrix}\right) \rightarrow (\epsilon) \quad [\textit{end}]$$

$$\left(\begin{smallmatrix} F \\ F \end{smallmatrix}\right) \rightarrow \left(\begin{smallmatrix} F \\ F \end{smallmatrix}\right) \circ \left(\begin{smallmatrix} T \\ T \end{smallmatrix}\right) \quad [\textit{iter}]$$

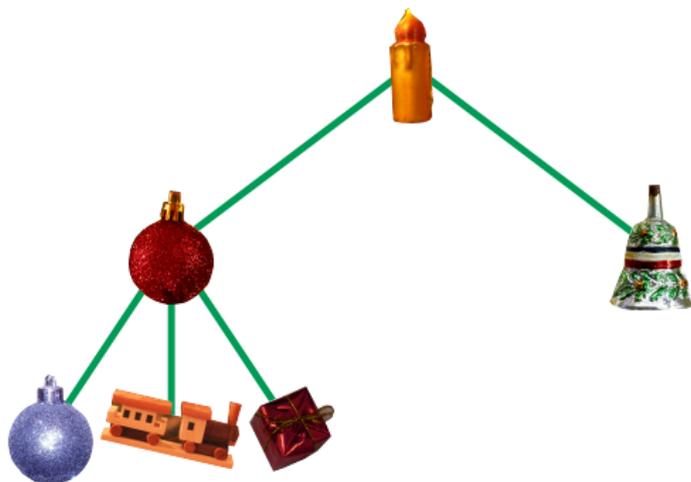
$$\left(\begin{smallmatrix} T \\ T \end{smallmatrix}\right) \rightarrow \left(\begin{smallmatrix} F \\ F \end{smallmatrix}\right) \left(\begin{smallmatrix} x \\ x \end{smallmatrix}\right) \quad [\textit{align}]$$

$$\left(\begin{smallmatrix} T \\ T \end{smallmatrix}\right) \rightarrow \left(\begin{smallmatrix} F \\ F \end{smallmatrix}\right) \left(\begin{smallmatrix} - \\ - \end{smallmatrix}\right) \quad [\textit{indel}]$$

$$\left(\begin{smallmatrix} T \\ T \end{smallmatrix}\right) \rightarrow \left(\begin{smallmatrix} F \\ F \end{smallmatrix}\right) \left(\begin{smallmatrix} x \\ - \end{smallmatrix}\right) \quad [\textit{delin}]$$

(1)

The index structure is based on **postorder** of trees.



Grammars

Tree Alignment

$$\left(\begin{matrix} S \\ S \end{matrix}\right) \rightarrow \left(\begin{matrix} F \\ F \end{matrix}\right) \quad [\textit{start}]$$

$$\left(\begin{matrix} F \\ F \end{matrix}\right) \rightarrow (\epsilon) \quad [\textit{end}]$$

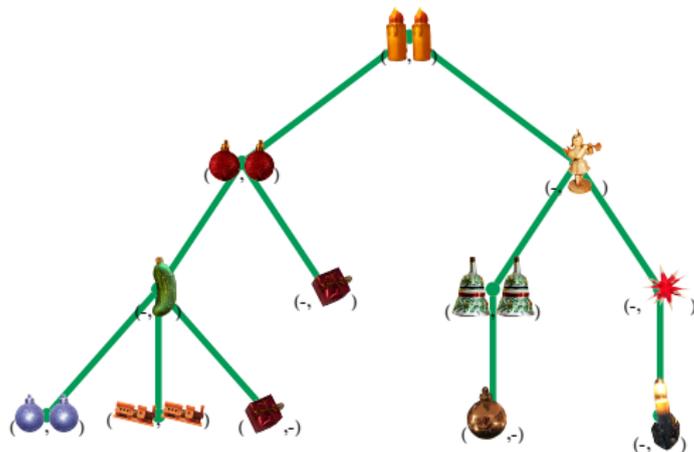
$$\left(\begin{matrix} F \\ F \end{matrix}\right) \rightarrow \left(\begin{matrix} T \\ T \end{matrix}\right) \circ \left(\begin{matrix} F \\ F \end{matrix}\right) \quad [\textit{iter}]$$

$$\left(\begin{matrix} T \\ T \end{matrix}\right) \rightarrow \left(\begin{matrix} n \\ n \end{matrix}\right) \left(\begin{matrix} F \\ F \end{matrix}\right) \quad [\textit{align}]$$

$$\left(\begin{matrix} T \\ T \end{matrix}\right) \rightarrow \left(\begin{matrix} - \\ n \end{matrix}\right) \left(\begin{matrix} F \\ F \end{matrix}\right) \quad [\textit{indel}]$$

$$\left(\begin{matrix} T \\ T \end{matrix}\right) \rightarrow \left(\begin{matrix} n \\ - \end{matrix}\right) \left(\begin{matrix} F \\ F \end{matrix}\right) \quad [\textit{delin}]$$

(1)



The index structure is based on **preorder** of trees.

Why?

- DP algorithms on trees:
 - ▶ small parsimony problem
 - ▶ phylogenetic targeting
 - ▶ tree editing
 - ▶ tree alignment (with affine gap costs)
- automatized DP on various data structures (in future)
- inside/outside: DP on probabilities



Challenge

Convert this title:

Why tree alignment doesn't have to suck

Challenge

Convert this title:

Why tree alignment doesn't have to suck

into something:

- referring to trees or forests
- but no other plants
- funny
- but not too funny ;)

The best submission will receive a *christmas cucumber!*



Acknowledgements



Thanks to...

- Peter Stadler
- Christian Höner zu Siederdisen

Acknowledgements



Thanks to...

- Peter Stadler
- Christian Höner zu Siederdisen

..and..

- Sophia for pictures,
- Rojin for planting trees,
- Marc for the story of 'Weihnachtsgurke'

Acknowledgements



Thanks to...

- Peter Stadler
- Christian Höner zu Siederdisen

..and..

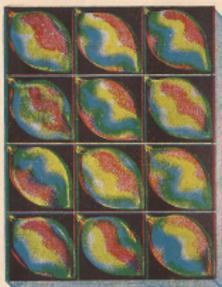
- Sophia for pictures,
- Rojin for planting trees,
- Marc for the story of 'Weihnachtsgurke'

Thank you for your attention!

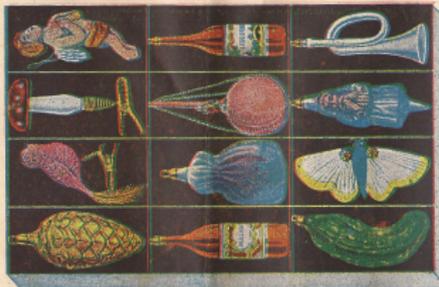
Proof for christmas cucumber!

LYRA-FAHRRAD-WERKE, HERMANN KLAASSEN, PRENZLAU

Erstes, ältestes, größtes und leistungsfähigstes Spezial-Haus für Fahrräder, Sportartikel, Uhren, Goldwaren, Waffen, Spielwaren



Nr. 5045. Christbaum schmuck
Ganz wunderbares Sortiment in allen Regenbogenfarben glänzendste Glasstücke aus neu!
Länge bis ca. 10 cm.
Karton mit 12 Stück 85 Pfg.



Nr. 5046. Christbaum schmuck
Dieses ganz reizende Sortiment aus Glaswaren eignet sich ganz besonders für kleine Kinder und hat im letzten Jahre ganz besonders großen Erfolg gefunden. Alles ist beweglich und kann auch als Spielzeug benutzt werden. Wirklich sehr schön.
Karton mit 12 Stück 85 Pfg.



Nr. 5047. Christbaum schmuck
Feinstes Sortiment in dem großartigen Farben spielend naturgetreu. Wirklich ganzes Zusammenhängen. Reizvolle Wirkung am Baum.
Größe ca. 10 cm.
Karton mit 12 Stück 95 Pfg.

