

Knuth-Plass Line Wrapping Algorithm

A. Xing

22 February 2016

Line wrapping

Where do we break lines to fit them to a page?

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face. Close by the king's castle lay a great dark forest, and under an old lime-tree in the forest was a well, and when the day was very warm, the king's child went out into the forest and sat down by the side of the cool fountain, and when she was bored she took a golden ball, and threw it up on high and caught it, and this ball was her favorite plaything.

Anatomy of a paragraph

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face.

Anatomy of a paragraph

box

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face.

Box Indivisible block - could be a character, syllable, etc.

Anatomy of a paragraph

In olden box times glue when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face.

- Box** Indivisible block - could be a character, syllable, etc.
- Glue** Boxes that can be stretched and shrunken - mostly whitespace

Anatomy of a paragraph

In olden box times when glue wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that penalty the sun itself, which has seen so much, was astonished whenever it shone in her face.

Box Indivisible block - could be a character, syllable, etc.

Glue Boxes that can be stretched and shrunken - mostly whitespace

Penalty “Optional” characters - mostly hyphens

More about glue

- **Stretchability** is the maximum ratio glues are allowed to expand
- **Shrinkability** is the opposite

More about glue

So the minimum width of a line is

$$width_{min} = width_{boxes} + num_{glue} \times width_{glue} \times shrinkability$$

And the maximum width is

$$width_{max} = width_{boxes} + num_{glue} \times width_{glue} \times stretchability$$

More about glue

So the minimum width of a line is

$$width_{min} = width_{boxes} + num_{glue} \times width_{glue} \times shrinkability$$

And the maximum width is

$$width_{max} = width_{boxes} + num_{glue} \times width_{glue} \times stretchability$$

This will help us determine where possible line breaks are

Boxes & glue

- Scan input and produce array of box and glue elements
- Break up words into syllables to allow hyphenation (by adding penalties)

Boxes & glue

- Scan input and produce array of box and glue elements
- Break up words into syllables to allow hyphenation (by adding penalties)
 - But that's another problem

Finding feasible line breaks

- Go through array of boxes and glue until you get to potential line break candidates
- “Potential” means the line broken at that point would fit in the page
- $width_{min} \leq width \leq width_{max}$

Active nodes

- We're going to store the line break candidates as nodes in a tree
- Start off with a single active node representing start of paragraph
- For each new element, see if it is a line break candidate with respect to all the currently active nodes

Active nodes



In olden times when wishing still helped one, there lived

a

king

whose daughters were all beautiful, but the youngest

was

so

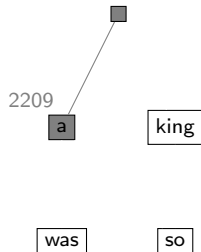
beautiful that the sun itself, which has seen so much

Active nodes

In olden times when wishing still helped one, there lived

whose daughters were all beautiful, but the youngest

beautiful that the sun itself, which has seen so much

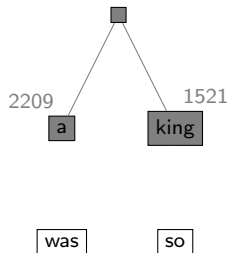


Active nodes

In olden times when wishing still helped one, there lived

whose daughters were all beautiful, but the youngest

beautiful that the sun itself, which has seen so much

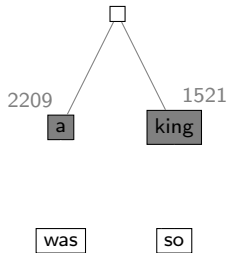


Active nodes

In olden times when wishing still helped one, there lived

whose daughters were all beautiful, but the youngest

beautiful that the sun itself, which has seen so much

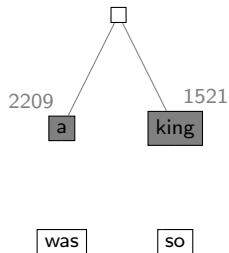


Active nodes

In olden times when wishing still helped one, there lived

whose daughters were all beautiful, but the youngest

beautiful that the sun itself, which has seen so much

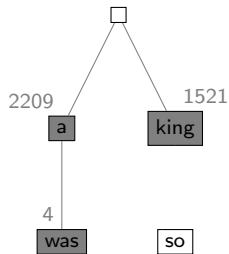


Active nodes

In olden times when wishing still helped one, there lived

whose daughters were all beautiful, but the youngest

beautiful that the sun itself, which has seen so much

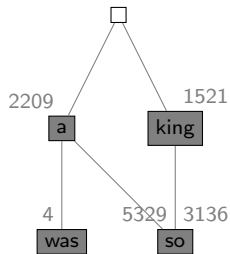


Active nodes

In olden times when wishing still helped one, there lived

whose daughters were all beautiful, but the youngest

beautiful that the sun itself, which has seen so much



Finding the optimal solution

- Now that we've figured out where all the *possible* breaks are, we need to find the optimal solution
- But first, we need a way to compare two possibilities in order to pick the better one

Interlude: Badness

- Introduce a measure of **badness**
- Represents how bad a line looks

Interlude: Badness

High badness:

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face

Low badness:

In olden times when wishing still helped one, there lived a king whose daughters were all beautiful, but the youngest was so beautiful that the sun itself, which has seen so much, was astonished whenever it shone in her face

Interlude: Badness

- Remember stretchability and shrinkability?
- Badness is the amount of stretching or shrinking
- If the line fits perfectly (no stretching/shrinking), the badness is 0

Interlude: Demerits

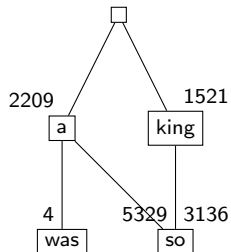
- The final “cost” of breaking a line at a particular point is calculated using **demerits**
- Often the square or cube of badness, plus some other parameters (e.g. penalties)
- Many different ways of calculating demerits

Interlude: Demerits

- The final “cost” of breaking a line at a particular point is calculated using **demerits**
- Often the square or cube of badness, plus some other parameters (e.g. penalties)
- Many different ways of calculating demerits
- But that’s another problem

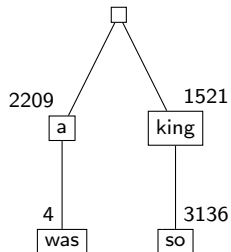
Finding the optimal solution

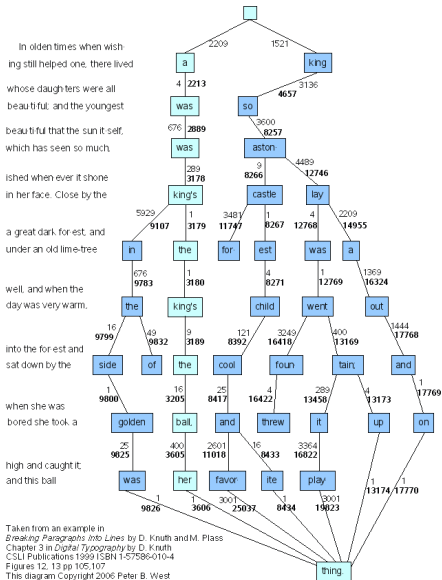
Only keep the best option when adding nodes to the tree



Finding the optimal solution

Only keep the best option when adding nodes to the tree





Taken from an example in *Breaking Paragraphs into Lines* by D. Knuth and M. Plass Chapter 3 in *Digital Typography* by D. Knuth CSLI Publications 1999 ISBN 1-57586-010-4 Figures 12, 13 pp 105, 107 This diagram Copyright 2006 Peter B. West

References

- Knuth, Donald E.; Plass, Michael F. (1981), “Breaking paragraphs into lines”, *Software: Practice and Experience* 11 (11): 11191184, doi:10.1002/spe.4380111102.
- West, Peter B.; Folio (2007), “Knuth & Plass line-breaking Revisited” ,
<http://defoe.sourceforge.net/folio/knuth-plass.html>