Pushdown compression

Pilar Albert, Elvira Mayordomo, Philippe Moser

Sylvain Perifel

LIP, ENS Lyon

Paris, May 30, 2008

Motivations

- New compression algorithms for structured documents (XML): behaviour depending on the current tag
  $\rightarrow$ use of a stack to push and pop tags.

Motivations

- New compression algorithms for structured documents (XML): behaviour depending on the current tag
  $\rightarrow$ use of a stack to push and pop tags.
- Very simple algorithms $\rightarrow$ pushdown automata.
- Easy to compress and decompress.
  Practical tests: better performances than zip (Lempel-Ziv).

Motivations

- New compression algorithms for structured documents (XML): behaviour depending on the current tag
  $\rightarrow$ use of a stack to push and pop tags.
- Very simple algorithms $\rightarrow$ pushdown automata.
- Easy to compress and decompress.
  Practical tests: better performances than zip (Lempel-Ziv).
- Need for a theoretical study.

Outline

1. Introduction (LZ, FS)
2. Pushdown compression
3. Pushdown beats LZ
4. LZ beats pushdown
5. Conclusion

Outline

1. Introduction (LZ, FS)
2. Pushdown compression
3. Pushdown beats LZ
4. LZ beats pushdown
5. Conclusion

Compression

- Lossless compression.

Compression

- Lossless compression.

- Compressor: injective and computable function $f : \{0, 1\}^* \to \{0, 1\}^*$.

Compression

- Lossless compression.
- Compressor: injective and computable function $f : \{0,1\}^* \to \{0,1\}^*$.
- Compression ratio on a finite word x:

$$\rho_f(x) = \frac{|f(x)|}{|x|}.$$

Compression ratio on an infinite sequence S:

$$\rho_f(S) = \limsup_{n\to\infty} \rho_f(S[1..n]).$$

Lempel-Ziv

Text to be compressed:

0 1 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 1

Compression result:

Lempel-Ziv

Text to be compressed:

0

ϵ/0 1 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 1

Compression result:

ϵ;

Lempel-Ziv

Text to be compressed:

0 1

ϵ/0/1 0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 1

Compression result:

$\epsilon;(0,0);$

Lempel-Ziv

Text to be compressed:

0 1 2

ϵ/0/1/0 0 0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 1

Compression result:

ϵ;$(0, 0)$;$(0, 1)$;

Lempel-Ziv

Text to be compressed:

0 1 2 3

ϵ/0/1/0 0/0 1 0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 1

Compression result:

ϵ;$(0, 0)$;$(0, 1)$;$(1, 0)$;

Lempel-Ziv

Text to be compressed:

0 1 2 3   4

ϵ/0/1/0 0/0 1/0 1 1 1 0 1 0 0 1 0 0 0 0 0 1 1 1

Compression result:

$\epsilon;(0,0);(0,1);(1,0);(1,1);$

Lempel-Ziv

Text to be compressed:

0 1 2 3   4   5

ϵ/0/1/0 0/0 1/0 1 1/1 0 1 0 0 1 0 0 0 0 0 1 1 1

Compression result:

$\epsilon;(0,0);(0,1);(1,0);(1,1);(4,1);$

Lempel-Ziv

Text to be compressed:

0 1 2 3 4 5 6 7 8 9 10

ϵ/0/1/0 0/0 1/0 1 1/1 0/1 0 0/1 0 0 0/0 0 1/1 1

Compression result:

ϵ;$(0,0)$;$(0,1)$;$(1,0)$;$(1,1)$;$(4,1)$;$(2,0)$;$(6,0)$;$(7,0)$;$(3,1)$;$(2,1)$

Lempel-Ziv

Text to be compressed:

0 1 2 3 4 5 6 7 8 9 10

ϵ/0/1/0 0/0 1/0 1 1/1 0/1 0 0/1 0 0 0/0 0 1/1 1

Compression result:

$\epsilon;(0,0);(0,1);(1,0);(1,1);(4,1);(2,0);(6,0);(7,0);(3,1);(2,1)$

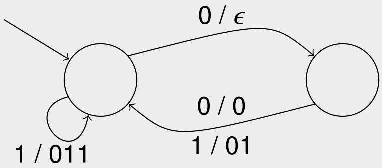
Lemma

- *If p is the number of phrases, then $|LZ(x)| = p \log p$.*
- *For all x, the compression ratio $\rho_{LZ}(x)$ satisfies*

$$\frac{\log |x|}{\sqrt{|x|}} \leq \rho_{LZ}(x) \leq 1 + o(1).$$

Finite-state compression (1)

Finite-state transducer: finite-state automaton that outputs letters at each transition
$\to$ function $f : \{0,1\}^* \to \{0,1\}^*$.

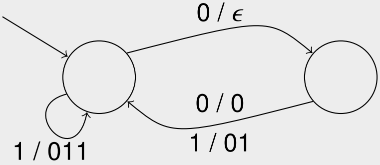


$$00 \mapsto 0, \quad 01 \mapsto 01, \quad 1 \mapsto 011.$$

Example: 00 00 01 1 1 00 $\mapsto$ 0 0 01 011 011 0.

Finite-state compression (1)

Finite-state transducer: finite-state automaton that outputs letters at each transition
→ function $f : \{0,1\}^* \to \{0,1\}^*$.



$$00 \mapsto 0, \quad 01 \mapsto 01, \quad 1 \mapsto 011.$$

Example: 00 00 01 1 1 00 $\mapsto$ 0 0 01 011 011 0.

Finite-state compressor: injective finite-state transducer (taking into account the final state).

Finite-state compression (2)

For a finite-state compressor C: compression ratio of an infinite sequence S

$$\rho_C(S) = \limsup_{n \to \infty} \frac{|C(S[1..n])|}{n}.$$

Finite-state compression (2)

For a finite-state compressor C: compression ratio of an infinite sequence S

$$\rho_C(S) = \limsup_{n\to\infty} \frac{|C(S[1..n])|}{n}.$$

Finite-state compression ratio:

$$\rho_{FS}(S) = \inf_{C \in FS} \rho_C(S).$$

LZ better than FS

Theorem (Lempel, Ziv, 1979)

On every infinite sequence $S \in \{0,1\}^{\mathbb{N}}$, Lempel-Ziv is better than any finite-state compressor, that is,

$$\rho_{LZ}(S) \leq \rho_{FS}(S).$$

Outline

Pushdown transducers

Pushdown compressor = finite-state transducer with a stack.

The transition is done according both to the symbol read and to the topmost symbol of the stack.

Each transition either pushes or pops symbols from the stack.

Pushdown transducers

Pushdown compressor = finite-state transducer with a stack.

The transition is done according both to the symbol read and to the topmost symbol of the stack.

Each transition either pushes or pops symbols from the stack.

PD compression ratio: $\rho_{PD}(S) = \inf_{C \in PD} \rho_C(S)$.

Pushdown transducers

Pushdown compressor = finite-state transducer with a stack.

The transition is done according both to the symbol read and to the topmost symbol of the stack.

Each transition either pushes or pops symbols from the stack.

PD compression ratio: $\rho_{PD}(S) = \inf_{C \in PD} \rho_C(S)$.

Two variants: with or without endmarkers
$\rightarrow C(x)$ or $C(x\#)$ (enables to empty the stack).

Example

Proposition

Let $S = 0^\infty$.

- *The compression ratio on* S *of a finite-state compressor with* k *states is* $\geq 1/k$.
- *There exists a pushdown compressor with* k *states whose compression ratio on* S *is* $\leq 1/k^2$ *(with endmarkers).*

Example

Proposition

Let $S = 0^\infty$.

- *The compression ratio on* S *of a* *finite-state* *compressor with* k *states is* $\geq 1/k$.
- *There exists a* *pushdown* *compressor with* k *states whose compression ratio on* S *is* $\leq 1/k^2$ *(with endmarkers).*

Proof.

1. Let C be a FS compressor with k states.

Then C must output at least one symbol every k letters.

Otherwise there would exist u such that for all $i_0 \leq i \leq i_0 + k$, all the $u[1..i]$ have the same image.

Since there are only k states, this contradicts injectivity.

Example

Proposition

Let $S = 0^\infty$.

- *The compression ratio on* S *of a finite-state compressor with* k *states is* $\geq 1/k$.
- *There exists a pushdown compressor with* k *states whose compression ratio on* S *is* $\leq 1/k^2$ *(with endmarkers).*

Proof.

2. Let C be the following pushdown compressor on input 0^n:
   - it pushes $0^{n/k}$ on the stack (by counting modulo k);
   - at the end it pops the stack and outputs one symbol every k (by counting modulo k).

□

Remarks

- Same result as FS for pushdown without endmarkers.

Remarks

- Same result as FS for pushdown without endmarkers.
- LZ on $S = 0^\infty$ has compression ratio $0 \ldots$

Remarks

- Same result as FS for pushdown without endmarkers.
- LZ on $S = 0^\infty$ has compression ratio $0\ldots$
- but FS also!

$$\rho_{FS}(S) = \inf_{C \in FS} \rho_C(S) \leq 1/k \text{ for all } k.$$

Outline

1. Introduction (LZ, FS)
2. Pushdown compression
3. Pushdown beats LZ
4. LZ beats pushdown
5. Conclusion

Pushdown beats LZ

Theorem

There exists a sequence S such that

$$\rho_{PD}(S) = 1/2 \text{ (without endmarkers)}$$

and

$$\rho_{LZ}(S) = 1.$$

The idea

Pushdown compresses palindromes with ratio $\simeq 1/2$...

but LZ not always.

The idea

Pushdown compresses palindromes with ratio $\simeq 1/2\ldots$

but LZ not always.

$\rightarrow$ build a sequence of the form

$$S = u_1 \bar{u}_1 u_2 \bar{u}_2 \ldots$$

with well-chosen words u_i (here $\bar{u}$ stands for the mirror of u).

Proof

Let $E_n \subset \{0,1\}^n$ be the set of words of size n that are not palindromes. Let $u_1, \ldots, u_{|E_n|/2}$ be $|E_n|/2$ words of E_n such that $\forall i, j, u_i \neq \bar{u}_j$. Then

$$u_1 \ldots u_{|E_n|/2} \ \bar{u}_{|E_n|/2} \ldots \bar{u}_1$$

is LZ-incompressible but 1/2-PD-compressible.
$\rightarrow$ repeat this for all sizes n to obtain the infinite sequence S. $\square$

Outline

LZ beats pushdown

Theorem

There exists a sequence S such that

$$\rho_{LZ}(S) = 0$$

and

$$\rho_{PD}(S) = 1 \text{ (with endmarkers).}$$

The idea

LZ compresses repetitions very well (ratio tends to 0). . .

but pushdown not always.

The idea

LZ compresses repetitions very well (ratio tends to 0)...

but pushdown not always.

- Show that some repetitions are not compressed by pushdown (→ pumping lemma);
- build a sequence of the form

$$S = u_1^{n_1} u_2^{n_2} \ldots$$

  for well chosen u_i and n_i.

LZ and repetitions

Lemma

Let u be a word.
The compression ratio of LZ on u^n is $O(\frac{\log n}{\sqrt{n}})$ (and thus tends to 0 when $n \to \infty$).

LZ and repetitions

Lemma

Let u be a word.
The compression ratio of LZ on u^n is $O(\frac{\log n}{\sqrt{n}})$ (and thus tends to 0 when $n \to \infty$).

Proof.
For all k, there are at most $|u|$ different words of size k in u^n.
Call p the number of phrases in the parsing of u^n by LZ algorithm.
Let t_k be the number of phrases of size k. We have:

$$|u^n| = \sum_{k \geq 1} t_k \geq \sum_{k=1}^{p/|u|} k|u| \geq \frac{p^2}{2|u|}.$$

Thus $p = O(\sqrt{n})$ and $|LZ(x)| = p \log p$. □

PD and repetitions

Let C be a pushdown compressor.

Suppose there is a pumping lemma: on input uv^nw, C has each time the same behaviour on v.

If v is not compressible, then $C(uv^nw) \geq n|v|$,
thus $\rho_C(uv^nw) \to 1$.

Pumping lemma

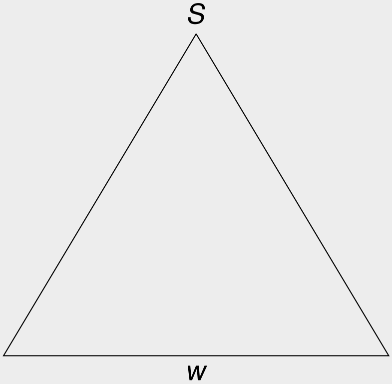
Theorem

Let A be a pushdown transducer (working without endmarkers). There exist two constants $\alpha, \beta > 0$ such that all word w can be cut in three pieces $w = tuv$ satisfying:

- $|u| \geq \lfloor \alpha |w|^{\beta} \rfloor$;
- *if* $C(tuv) = xyz$ *then* $C(tu^n) = xy^n$.

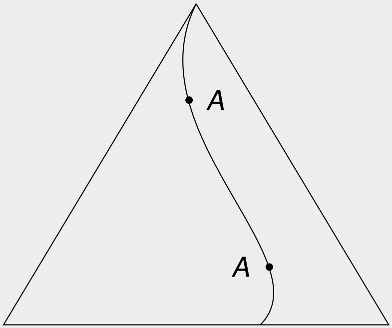
Acceptors: reminder

Equivalence pushdown automata / context-free grammars.



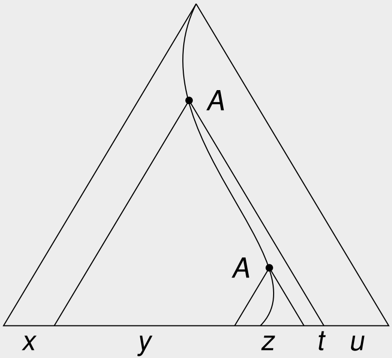
Acceptors: reminder

Equivalence pushdown automata / context-free grammars.



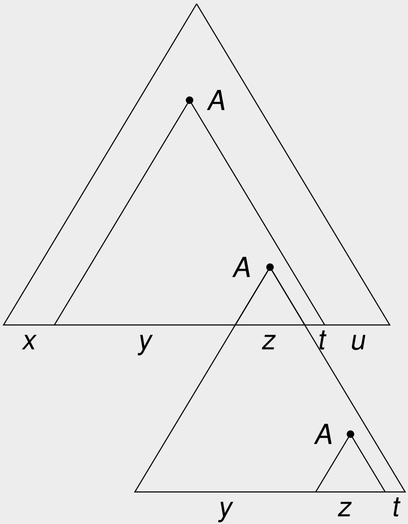
Acceptors: reminder

Equivalence pushdown automata / context-free grammars.



Acceptors: reminder

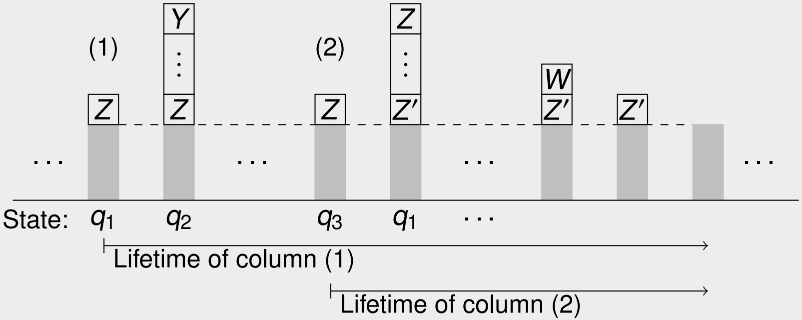
Equivalence pushdown automata / context-free grammars.



Transducers: proof (1)

Transducers: proof (1)

- Lifetime of a column: never go below its top symbol.
- Equivalent columns: c' in the lifetime of c and same state/top symbol.

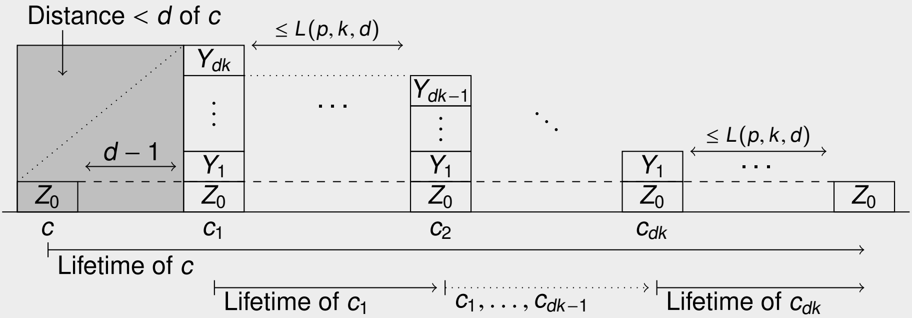


Transducers: proof (2)

- p: number of pairs state/top symbol;
- k: max number of symbols pushed by one rule;
- $L(p, k, d)$: maximum lifetime of a column during which no pair of equivalent columns are at distance $\geq d$.

Transducers: proof (2)

- p: number of pairs state/top symbol;
- k: max number of symbols pushed by one rule;
- $L(p,k,d)$: maximum lifetime of a column during which no pair of equivalent columns are at distance $\geq d$.
- $L(p+1,k,d) = d + kdL(p,k,d)$.



The endmarker

Theorem

Let A be a pushdown transducer (working with endmarkers). There exist two constants $\alpha, \beta > 0$ such that all word w can be cut in three pieces $w = tuv$ satisfying:

- $|u| \geq \lfloor \alpha |w|^{\beta} \rfloor$;
- *there are five words x, x', y, y', z such that* $C(tu^n v\#) = xy^n zy'^n x'$.

Remark.

The same is true with an initially nonempty stack.

LZ beats PD

Theorem

There exists a sequence S such that

$$\rho_{LZ}(S) = 0 \text{ and } \rho_{PD}(S) = 1 \text{ (with endmarkers).}$$

LZ beats PD

Theorem

There exists a sequence S such that

$$\rho_{LZ}(S) = 0 \text{ and } \rho_{PD}(S) = 1 \text{ (with endmarkers).}$$

Proof.
Let w_i be a sufficiently big Kolmogorov-random word
$\rightarrow$ cut it in three pieces $w_i = t_i u_i v_i$, with u_i big enough (thus incompressible), according to the i-th pushdown transducers. Then

$$S = t_1 u_1^{n_1} t_2 u_2^{n_2} \dots$$

(for sufficiently large integers n_i) is LZ-compressible but not PD-compressible. □

Outline

1. Introduction (LZ, FS)
2. Pushdown compression
3. Pushdown beats LZ
4. LZ beats pushdown
5. Conclusion

Summary

- Introduction of pushdown compression.
- Strictly better than finite-state compression.

Summary

- Introduction of pushdown compression.
- Strictly better than finite-state compression.
- Better than Lempel-Ziv on some sequences (palindromes: compression ratio 1/2 instead of 1).

Summary

- Introduction of pushdown compression.
- Strictly better than finite-state compression.
- Better than Lempel-Ziv on some sequences (palindromes: compression ratio 1/2 instead of 1).
- Worse than Lempel-Ziv on some sequences (repetitions: compression ratio 1 instead of 0).

Future work

- Lower bound on the compression ratio of a PD compressor with k states with endmarkers?
- Better separation for "PD beats LZ"?

Outline

1. Introduction (LZ, FS)
2. Pushdown compression
3. Pushdown beats LZ
4. LZ beats pushdown
5. Conclusion