Outline

Outline

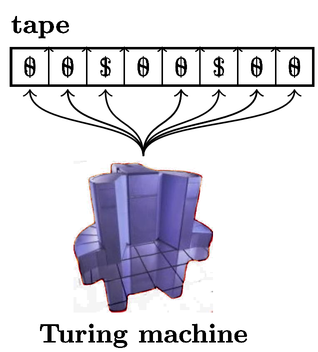
Contents

Part I
Main Talk

1 The Model of Overhead-Free Computation

1.1 The Standard Model of Linear Space

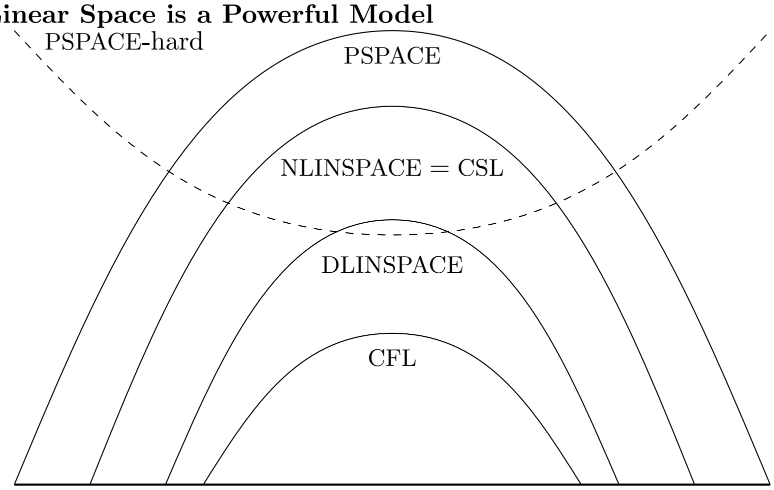
The Standard Model of Linear Space



Characteristics

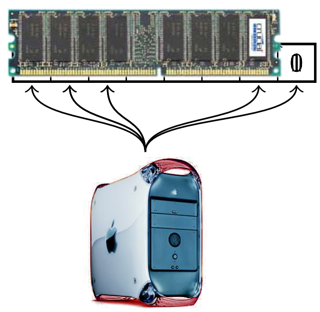
- Input fills *fixed-size tape*
- Input may be *modified*
- Tape alphabet *is larger than* input alphabet

Linear Space is a Powerful Model



1.2 Our Model of Absolutely No Space Overhead

Our Model of "Absolutely No Space Overhead"



Turing machine

Characteristics

- Input fills *fixed-size tape*
- Input may be *modified*
- Tape alphabet *equals* input alphabet

Intuition

- Tape is used like a RAM module.

Definition of Overhead-Free Computations

Definition
A Turing machine is *overhead-free* if

- it has only a single tape,
- writes only on input cells,
- writes only symbols drawn from the input alphabet.

Overhead-Free Computation Complexity Classes

Definition
A language $L \subseteq \Sigma^*$ is in

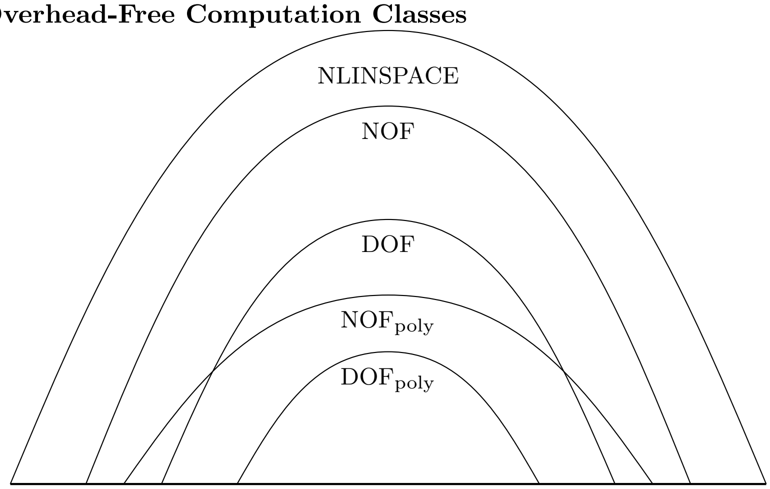
DOF if L is accepted by a deterministic overhead-free machine with input alphabet Σ,

DOF$_{\text{poly}}$ if L is accepted by a deterministic overhead-free machine with input alphabet Σ in polynomial time.

NOF is the nondeterministic version of DOF,

NOF$_{\text{poly}}$ is the nondeterministic version of DOF_{poly}.

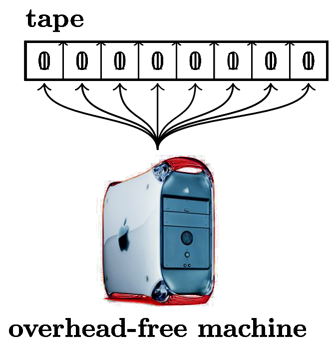
Simple Relationships among Overhead-Free Computation Classes



2 The Power of Overhead-Free Computation

2.1 Palindromes

Palindromes Can be Accepted in an Overhead-Free Way

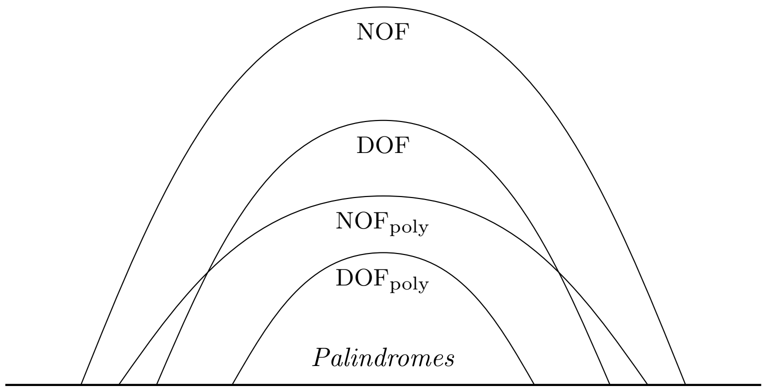


Algorithm

Phase 1:
Compare first and last bit
Place left end marker
Place right end marker

Phase 2:
Compare bits next to end markers
Find left end marker
Advance left end marker
Find right end marker
Advance right end marker

Relationships among Overhead-Free Computation Classes



2.2 Linear Languages

A Review of Linear Grammars

Definition
A grammar is *linear* if it is context-free and
there is only one nonterminal per right-hand side.

Example
$G_1\colon S \to 00S0 \mid 1$ and $G_2\colon S \to 0S10 \mid 0$.

Definition
A grammar is *deterministic* if
"there is always only one rule that can be applied."

Example
$G_1\colon S \to 00S0 \mid 1$ is deterministic.
$G_2\colon S \to 0S10 \mid 0$ is *not* deterministic.

Deterministic Linear Languages
Can Be Accepted in an Overhead-Free Way

Theorem
Every deterministic linear language is in $\mathrm{DOF}_{\mathrm{poly}}$.

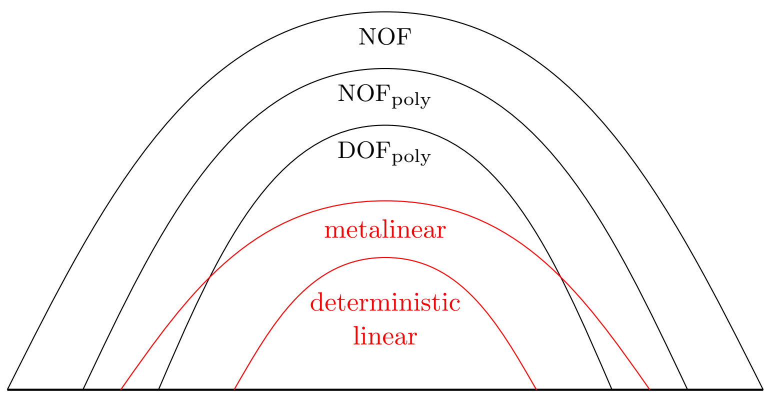
Metalinear Languages
Can Be Accepted in an Overhead-Free Way

Definition
A language is *metalinear* if it is the concatenation of linear languages.

Example
TRIPLE-PALINDROME $= \{uvw \mid u, v, \text{ and } w \text{ are palindromes}\}$.

Theorem
Every metalinear language is in $\mathrm{NOF}_{\mathrm{poly}}$.

Relationships among Overhead-Free Computation Classes



2.3 Context-Free Languages with a Forbidden Subword

Definition of Almost-Overhead-Free Computations

Definition
A Turing machine is *almost-overhead-free* if

- it has only a single tape,
- writes only on input cells,
- writes only symbols drawn from the input alphabet *plus one special symbol.*

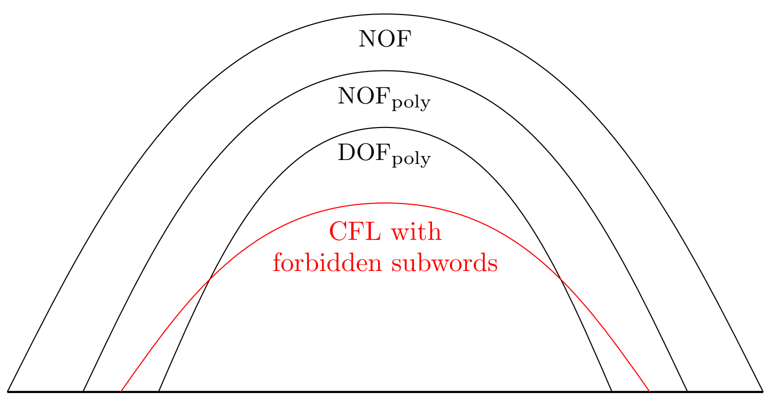
Context-Free Languages with a Forbidden Subword Can Be Accepted in an Overhead-Free Way

Theorem
Let L be a context-free language with a forbidden word.
Then $L \in \mathrm{NOF}_{\mathrm{poly}}$.

Skip proof

Proof.
Every context-free language can be accepted by a nondeterministic almost-overhead-free machine in polynomial time.

Relationships among Overhead-Free Computation Classes



2.4 Languages Complete for Polynomial Space

Overhead-Free Languages can be PSPACE-Complete

Theorem
DOF contains languages that are complete for PSPACE.

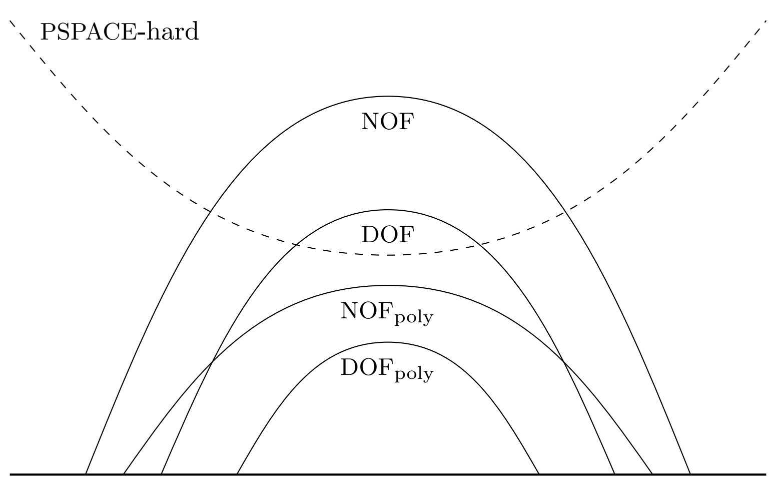
pspacecomplete¡2¿Proof details

Proof.

- Let $A \in$ DLINSPACE be PSPACE-complete.
  Such languages are known to exist.
- Let M be a linear space machine that accepts $A \subseteq \{0,1\}^*$ with tape alphabet Γ.
- Let $h\colon \Gamma \to \{0,1\}^*$ be an isometric, injective homomorphism.
- Then $h(L)$ is in DOF and it is PSPACE-complete.

pspacecomplete¡1¿Return

Relationships among Overhead-Free Computation Classes



3 Limitations of Overhead-Free Computation

3.1 Linear Space is Strictly More Powerful

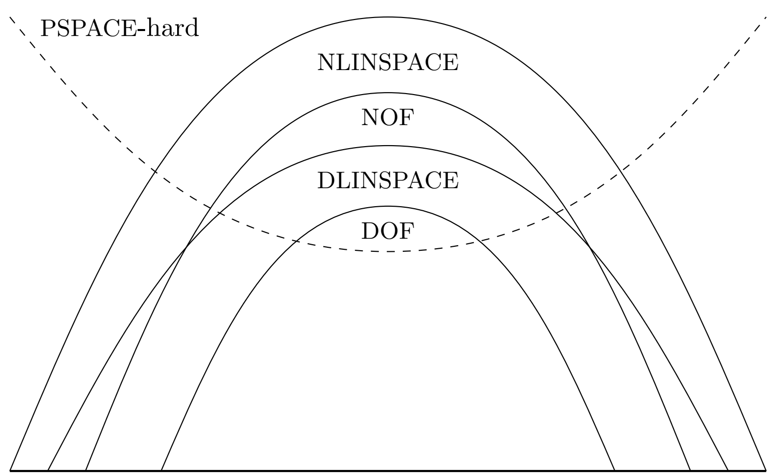
Some Context-Sensitive Languages
Cannot be Accepted in an Overhead-Free Way

Theorem
DOF $\subsetneq$ DLINSPACE.

Theorem
NOF $\subsetneq$ NLINSPACE.

The proofs are based on old diagonalisations due to Feldman, Owings, and Seiferas.

Relationships among Overhead-Free Computation Classes



Candidates for Languages that
Cannot be Accepted in an Overhead-Free Way

Conjecture
DOUBLE-PALINDROMES $\notin$ DOF.

Conjecture
$\{ww \mid w \in \{0,1\}^*\} \notin$ NOF.

Proving the first conjecture would show DOF $\subsetneq$ NOF.

Summary

Summary

Summary

- Overhead-free computation is a more faithful
  model of fixed-size memory.
- Overhead-free computation is *less powerful* than linear space.
- *Many* context-free languages can be accepted
  by overhead-free machines.
- We conjecture that *all* context-free languages are in $\mathrm{NOF}_{\mathrm{poly}}$.
- Our results can be seen as new results on the power of
  linear bounded automata with fixed alphabet size.

Further Reading

For Further Reading

References

[1] A. Salomaa. *Formal Languages.* Academic Press, 1973.

[2] E. Dijkstra. Smoothsort, an alternative for sorting in situ. *Science of Computer Programming*, 1(3):223–233, 1982.

[3] E. Feldman and J. Owings, Jr. A class of universal linear bounded automata. *Information Sciences*, 6:187–190, 1973.

[4] P. Jančar, F. Mráz, M. Plátek, and J. Vogel. Restarting automata. *FCT Conference 1995*, LNCS 985, pages 282–292. 1995.

A Appendix

Appendix Outline

Contents

A.1 Complete Languages

A.2 Improvements for Context-Free Languages

Improvements

Theorem

1. $\mathrm{DCFL} \subseteq \mathrm{DOF}_{\mathrm{poly}}$.
2. $\mathrm{CFL} \subseteq \mathrm{NOF}_{\mathrm{poly}}$.

A.3 Abbreviations

Explanation of Different Abbreviations