Computation with Absolutely No Space Overhead

Lane Hemaspaandra[1]   Proshanto Mukherji[1]   Till Tantau[2]

[1]Department of Computer Science
University of Rochester

[2]Fakultät für Elektrotechnik und Informatik
Technical University of Berlin

Developments in Language Theory Conference, 2003

Part I: Review of Last Lecture

Part II: Today's Lecture

The Model of Overhead-Free Computation
The Standard Model of Linear Space
Our Model of Absolutely No Space Overhead

The Power of Overhead-Free Computation
Palindromes
Linear Languages
Context-Free Languages with a Forbidden Subword
Languages Complete for Polynomial Space

Limitations of Overhead-Free Computation
Linear Space is Strictly More Powerful

Part II: Today's Lecture

The Model of Overhead-Free Computation
The Standard Model of Linear Space
Our Model of Absolutely No Space Overhead

The Power of Overhead-Free Computation
Palindromes
Linear Languages
Context-Free Languages with a Forbidden Subword
Languages Complete for Polynomial Space

Limitations of Overhead-Free Computation
Linear Space is Strictly More Powerful

Part II: Today's Lecture

The Model of Overhead-Free Computation
  The Standard Model of Linear Space
  Our Model of Absolutely No Space Overhead

The Power of Overhead-Free Computation
  Palindromes
  Linear Languages
  Context-Free Languages with a Forbidden Subword
  Languages Complete for Polynomial Space

Limitations of Overhead-Free Computation
  Linear Space is Strictly More Powerful

Part I

Review of Last Lecture

Some results.

Some results.

Models
Power of the Model
Limitations of the Model
Summary

Part II

Today's Lecture

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

0	0	1	0	0	1	0	0

Characteristics

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

Turing machine

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

$	0	1	0	0	1	0	0

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

\$	0	1	0	0	1	0	0

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

$	0	1	0	0	1	0	$

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

\$	0	1	0	0	1	0	\$

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

$	$	1	0	0	1	0	$

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

$	$	1	0	0	1	0	$

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

$	$	1	0	0	1	$	$

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

$	$	$	$	$	$	$	$

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

The Standard Model of Linear Space

tape

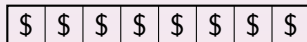| \$	\$	\$	\$	\$	\$	\$	\$

Characteristics

- Input fills fixed-size tape

- Input may be modified

- Tape alphabet is larger than input alphabet

Turing machine

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Linear Space is a Powerful Model

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Our Model of "Absolutely No Space Overhead"

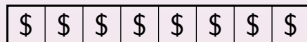tape

0	0	1	0	0	1	0	0

Turing machine

Characteristics

- Input fills fixed-size tape

- Input may be modified

- Tape alphabet equals input alphabet

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Our Model of "Absolutely No Space Overhead"

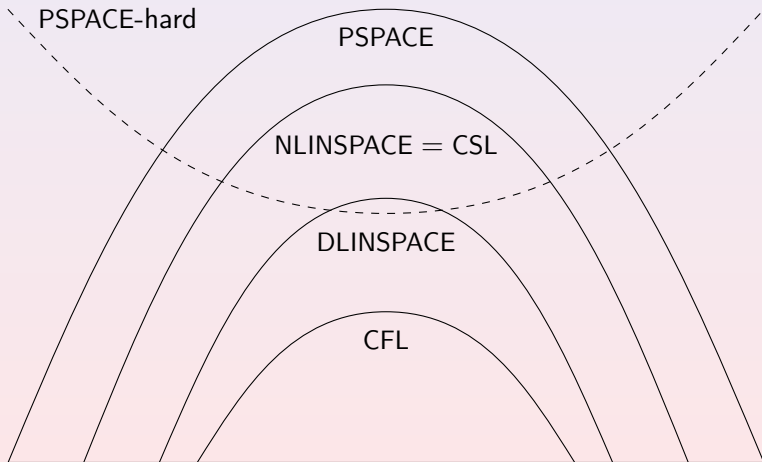tape

1	0	1	0	0	1	0	0

Turing machine

Characteristics

- Input fills fixed-size tape

- Input may be modified

- Tape alphabet equals input alphabet

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Our Model of "Absolutely No Space Overhead"

tape

1	0	1	0	0	1	0	0

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Our Model of "Absolutely No Space Overhead"

tape

1	0	1	0	0	1	0	1

Turing machine

Characteristics

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

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Our Model of "Absolutely No Space Overhead"

tape

1	0	1	0	0	1	0	1

Turing machine

Characteristics

- Input fills fixed-size tape

- Input may be modified

- Tape alphabet equals input alphabet

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Our Model of "Absolutely No Space Overhead"

tape

1	1	1	0	0	1	0	1

Characteristics

- Input fills fixed-size tape

- Input may be modified

- Tape alphabet equals
  input alphabet

Turing machine

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Our Model of "Absolutely No Space Overhead"

Intuition

- Tape is used like a RAM module.

Turing machine

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

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.

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

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_{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_{poly} is the nondeterministic version of DOF_{poly}.

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

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_{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_{poly} is the nondeterministic version of DOF_{poly}.

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

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$_{poly}$ 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$_{poly}$ is the nondeterministic version of DOF$_{poly}$.

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Overhead-Free Computation Complexity Classes

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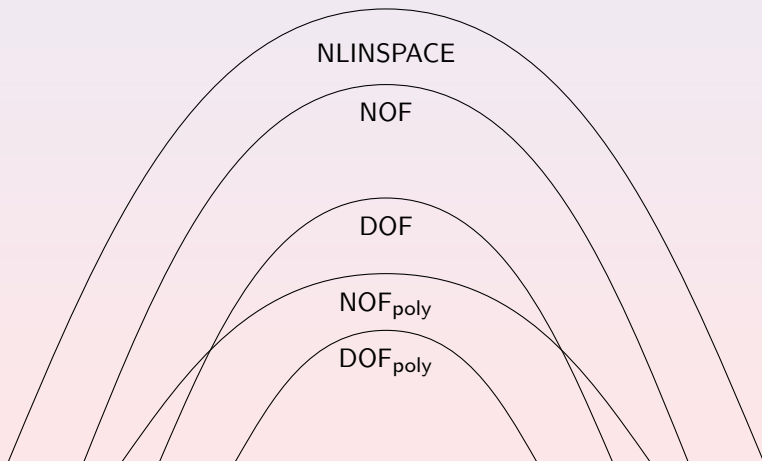
A language $L \subseteq \Sigma^*$ is in

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

DOF_{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_{poly} is the nondeterministic version of DOF_{poly}.

Models
Power of the Model
Limitations of the Model
Summary

Standard Model
Our Model

Simple Relationships among
Overhead-Free Computation Classes

NLINSPACE

NOF

DOF

NOF$_{poly}$

DOF$_{poly}$

Models   **Palindromes**
Power of the Model   **Linear Languages**
Limitations of the Model   **Forbidden Subword**
Summary   **Complete Languages**

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

1	0	1	0	0	1	0	0

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

Algorithm

tape

| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

Algorithm

tape

| 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 |

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

Algorithm

tape

| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

| 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 |

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

| 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

Algorithm

tape

0	1	1	0	0	1	1	0

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

0	0	1	0	0	1	1	0

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

0	0	1	0	0	1	1	0

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

0	0	1	0	0	1	0	0

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

tape

0	0	0	1	0	1	0	0

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

Algorithm

tape

0	0	0	1	0	1	0	0

overhead-free machine

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Palindromes Can be Accepted in an Overhead-Free Way

Algorithm

tape

| 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 |

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

overhead-free machine

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Relationships among Overhead-Free Computation Classes

NOF

DOF

NOF_{poly}

DOF_{poly}

Palindromes

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete 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: $S \rightarrow 00S0 \mid 1$.
G_2: $S \rightarrow 0S10 \mid 0$.

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

Example
G_1 is deterministic.
G_2 is not deterministic.

Models | Palindromes
Power of the Model | **Linear Languages**
Limitations of the Model | Forbidden Subword
Summary | Complete 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
$G_1 : S \rightarrow 00S0 \mid 1.$
$G_2 : S \rightarrow 0S10 \mid 0.$

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

Example
G_1 is deterministic.
G_2 is not deterministic.

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

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

Theorem
Every deterministic linear language is in DOF_{poly}.

Models | Palindromes
Power of the Model | **Linear Languages**
Limitations of the Model | Forbidden Subword
Summary | Complete Languages

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 NOF_{poly}.

Models | Palindromes
Power of the Model | **Linear Languages**
Limitations of the Model | Forbidden Subword
Summary | Complete Languages

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 NOF_{poly}.

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Metalinear Languages
Can Be Accepted in an Overhead-Free Way

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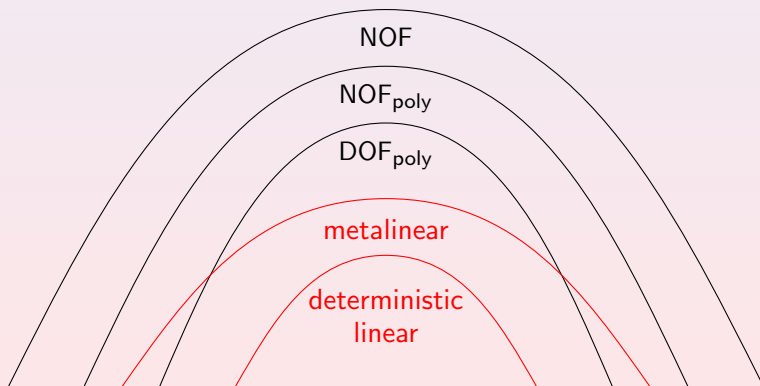A language is metalinear if it is the concatenation
of linear languages.

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

Theorem
Every metalinear language is in NOF_{poly}.

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Relationships among Overhead-Free Computation Classes

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

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.

Models   Palindromes
Power of the Model   Linear Languages
Limitations of the Model   **Forbidden Subword**
Summary   Complete Languages

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.

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

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.

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

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 $L \in \text{NOF}_{\text{poly}}$.

▸ Skip proof

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Context-Free Languages with a Forbidden Subword Can Be Accepted in an Overhead-Free Way
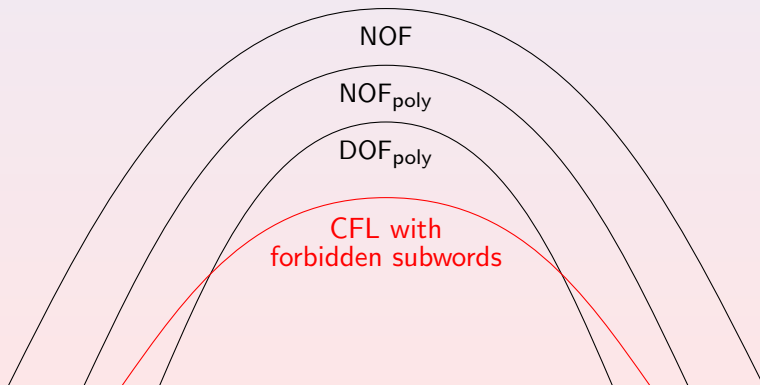
Theorem
Let L be a context-free language with a forbidden word.
Then $L \in \text{NOF}_{\text{poly}}$.

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

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Relationships among Overhead-Free Computation Classes

NOF

NOF_{poly}

DOF_{poly}

CFL with
forbidden subwords

Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Some PSPACE-complete Languages
Can Be Accepted in an Overhead-Free Way

Theorem
DOF contains languages that are complete for PSPACE.

▸ Proof details
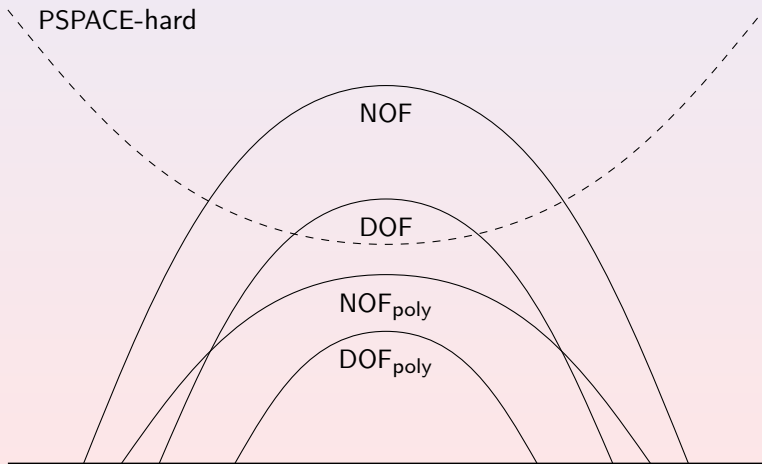
Models
Power of the Model
Limitations of the Model
Summary

Palindromes
Linear Languages
Forbidden Subword
Complete Languages

Relationships among Overhead-Free Computation Classes

Models
Power of the Model
Limitations of the Model
Summary

Strict Inclusion

Models
Power of the Model
Limitations of the Model
Summary

Strict Inclusion

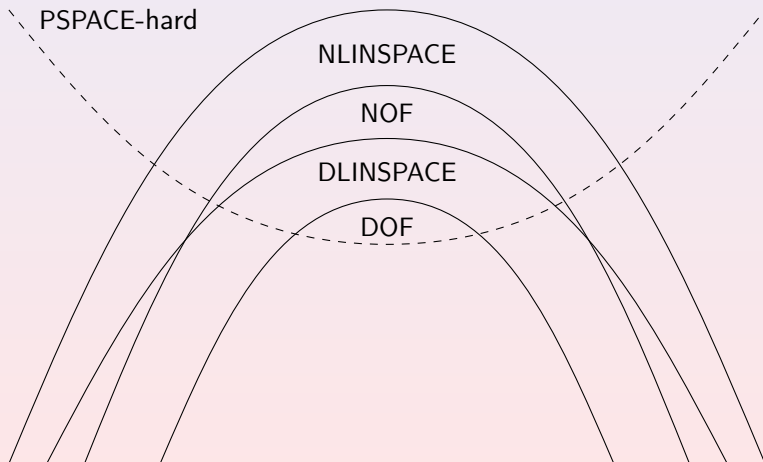# Some Context-Sensitive Languages
Cannot be Accepted in an Overhead-Free Way

Theorem
DOF $\subsetneq$ DLINSPACE.

Theorem
NOF $\subsetneq$ NLINSPACE.

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

Models
Power of the Model
Limitations of the Model
Summary

Strict Inclusion

Relationships among Overhead-Free Computation Classes

Models
Power of the Model
Limitations of the Model
Summary

Strict Inclusion

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.

Models
Power of the Model
Limitations of the Model
Summary

Summary
Further Reading

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 NOF_{poly}.

- Our results can be seen as new results on the power of linear bounded automata with fixed alphabet size.

Models
Power of the Model
Limitations of the Model
Summary

Summary
Further Reading

For Further Reading

📕 A. Salomaa.
Formal Languages.
Academic Press, 1973.

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

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

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

Models
Power of the Model
Limitations of the Model
Summary

Summary
Further Reading

For Further Reading

📕 A. Salomaa.
Formal Languages.
Academic Press, 1973.

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

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

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

Models
Power of the Model
Limitations of the Model
Summary

Summary
Further Reading

For Further Reading

📕 A. Salomaa.
Formal Languages.
Academic Press, 1973.

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

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

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

Models
Power of the Model
Limitations of the Model
Summary

Summary
Further Reading

For Further Reading

📕 A. Salomaa.
Formal Languages.
Academic Press, 1973.

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

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

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

Appendix

Overhead-Free Languages can be PSPACE-Complete

Theorem
DOF contains languages that are complete for PSPACE.

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.

Improvements

Theorem

1. $DCFL \subseteq DOF_{poly}$.

2. $CFL \subseteq NOF_{poly}$.

Explanation of Different Abbreviations

DOF	Deterministic Overhead-Free.
NOF	Nondeterministic Overhead-Free.
DOF_{poly}	Deterministic Overhead-Free, polynomial time.
DOF_{poly}	Nondeterministic Overhead-Free, polynomial time.

Table: Explanation of what different abbreviations mean.