A General BSS Model over Arbitrary Structures

Christine Gaßner
Greifswald

Cape Town
February, 2011
A general BSS model over arbitrary structures

1. The classical register machines and the BSS model
2. Comparison of BSS model and RAM’s
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s
   - Turing machines $\leftrightarrow$ BSS machines
   - Type-2 machines $\leftrightarrow$ RAM’s
   - Consequences
4. The Halting problems for several types of machines
5. Remarks to non-deterministic machines
1. The classical register machines and the BSS model

Some known models of computation

- **Turing machines** over \( \{0, 1\} \) (and a blank symbol)
  - Type-1
  - Type-2

  Meaning:
  - theory of computability and theory of complexity

- **Register machines** over \( \{a_1, \ldots, a_k\}, \mathbb{N}, \mathbb{R}, \ldots \)
  - Finite dimensional machines
  - Infinite dimensional machines
    - Offline
    - Online

  Meaning:
  - simple models for existing computers and theory of complexity

gassnerc@uni-greifswald.de
1. The classical register machines and the BSS model

Some known models of computation offline / online

Minsky (1961), Scott (1967), Blum/Shub/Smale (1989),... (offline):

\[ x_1, \ldots, x_n \rightarrow \text{Input procedure} \rightarrow \text{A program} \rightarrow \text{Output procedure} \rightarrow f(x) = y_1, \ldots, y_m \]
## Some known models of computation

### Offline / Online

<table>
<thead>
<tr>
<th>Model</th>
<th>Input Procedure</th>
<th>Program Execution Until Stop Criterion</th>
<th>Output Procedure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minsky (1961), Scott (1967), Blum/Shub/Smale (1989),... (offline):</td>
<td>( x ) ((x_1, \ldots, x_n) )</td>
<td>A program executed until stop criterion</td>
<td>( f(x) ) ((y_1, \ldots, y_m) )</td>
</tr>
<tr>
<td>Cook/Reckhow (1973), Aho/Hopcroft/Ullman (1974),... (offline / online):</td>
<td>((x_1, x_2, \ldots, x_n[, \ldots])) suitable inputs or codes of objects</td>
<td>A program executed until stop criterion</td>
<td>((y_1, y_2, \ldots, y_m[, \ldots]))</td>
</tr>
</tbody>
</table>

---

1. The classical register machines and the BSS model

---

[Email: gassnerc@uni-greifswald.de]
1. The classical register machines and the BSS model

Some known models of computation

offline / online

Minsky (1961), Scott (1967), Blum/Shub/Smale (1989),... (offline):

\[ x \rightarrow \text{Input procedure} \rightarrow \text{A program} \rightarrow \text{Output procedure} \rightarrow f(x) \]

Cook/Reckhow (1973), Aho/Hopcroft/Ullman (1974),... (offline / online):

\[ (x_1, x_2, ..., x_n[, ...]) \rightarrow \text{READ} \rightarrow \text{A program} \rightarrow \text{PRINT} \rightarrow (y_1, y_2, ..., y_m[, ...]) \]

suitable inputs or codes of objects

Weihrauch (1985?),... (online):

\[ (x_1, x_2, x_3, ...) \rightarrow \text{READ} \rightarrow \text{A program} \rightarrow \text{PRINT} \rightarrow (y_1, y_2, y_3, ...) \]

code/name of \( u \)

\[ u \rightarrow \text{code/name of } f(u) \rightarrow f(u) \]
Our goal

Definition of a general model of computation

- by
  - comparing known standard models
  - generalizing the main concepts

- in order to
  - unify known models
  - gain new insights in the relationship between known models (similarities and differences)
  - transfer results from one theory of computation to another
  - better understand the open problems in the classical theory of complexity (Why is it difficult to solve the classical P-NP problem? ...)

gassnerc@uni-greifswald.de
Our register machines

- **Registers**
  - a countable number of registers for storing numbers

- **Instructions**
  - start and halt instructions
  - operation and test instructions

- **Program**
  - a finite sequence of labelled instructions (M. L. Minsky, 1961)
  - represented by flow diagrams (Z. A. Melzak, 1961; J. Lambek, 1961)
  - a program schema with operation and predicate symbols (D. Scott, 1967)

- **Machine**
  - provides the interpretation for a program (D. Scott, 1967)

- **Model of computation**
  - the input and the output procedures
  - the machine
1. The classical register machines and the BSS model

Our register machines

- **Finite dimensional machines**
  - a finite number of registers: \( Z_1, \ldots, Z_m \)
  - Each configuration is determined by the label of the current instruction and the values of the registers.

- **Infinite dimensional machines**
  - an infinite number of registers: \( Z_1, Z_2, \ldots, Z_m, Z_{m+1}, Z_{m+2}, \ldots \)
  - \( m \) processor registers (or one accumulator)
  - without indirect addressing
  - with indirect addressing
    (can be realized by index registers \( I_1, \ldots, I_k \))

- gassnerc@uni-greifswald.de
1. The classical register machines and the BSS model

Examples for several structures

\[ \mathbb{Z}_2 = (\{0, 1\}; 0, 1; +, \cdot; =) \]
\[ \Rightarrow \text{Turing machine} \]
\[ \Rightarrow \text{Type-2 machines} \]

\[ \mathbb{N}_0 = (\mathbb{N}; 0; f_1, f_2; r_1), \quad f_1(n) = n + 1, \quad f_2(n) = n \div 1, \quad r_1(n) = \text{true iff } n \neq 0 \]
\[ \Rightarrow \text{counter machine} \]

\[ \mathbb{N} = (\mathbb{N}; \mathbb{N}; +, -; r_1) \]
\[ \Rightarrow \text{classical RAM} \]

\[ \mathbb{R} = (\mathbb{R}; \mathbb{R}; +, -; \cdot; \leq) \]
\[ \Rightarrow \text{real RAM} \]
\[ \Rightarrow \text{BSS model} \]
1. The classical register machines and the BSS model

**Instructions over \( \Sigma \)**

**A structure:** \( \Sigma = (U; c_1, c_2, \ldots; f_1, f_2, \ldots; r_1, r_2, \ldots) \)

**Computation:**

\[ l: Z_k := f_j(Z_{k1}, \ldots, Z_{km_j}); \]
\[ l: Z_k := c_j; \]

**Branching:**

\[ l: \text{if } r_j(Z_{k1}, \ldots, Z_{kn_j}) \text{ then goto } l_1 \text{ else goto } l_2; \]

**Copy:**

\[ l: Z_{lk} := Z_{lj}; \]

**Index computation:**

\[ I_k := 1; \quad I_k := I_k + 1; \quad \text{if } I_k = I_j \text{ then goto } l_1 \text{ else goto } l_2; \]
1. The classical register machines and the BSS model

### An example

#### for a finite dimensional machine

By Minsky (1961).

\[ \Sigma = (\mathbb{N}; 0; f_1, f_2; r_1), \quad f_1(n) = n + 1, \quad f_2(n) = n \div 1 \quad (0 \div 1 = 0), \quad r_1(n) = \text{true} \quad \text{iff} \quad n \neq 0 \]

**Registers:** \( Z_1, Z_2 \)

**Computation:**

\[
\begin{align*}
\text{l: } & Z_1 := Z_1 + 1; \quad \text{l: } Z_1 := Z_2 + 1; \quad \text{l: } Z_2 := Z_1 + 1; \quad \text{l: } Z_2 := Z_2 + 1; \\
\text{l: } & Z_1 := Z_1 \div 1; \quad \text{l: } Z_1 := Z_2 \div 1; \quad \text{l: } Z_2 := Z_1 \div 1; \quad \text{l: } Z_2 := Z_2 \div 1;
\end{align*}
\]

**Branching:**

\[
\begin{align*}
\text{l: } & \text{if } Z_1 \neq 0 \text{ then goto } l_1 \text{ else goto } l_2; \quad \text{l: } \text{if } Z_2 \neq 0 \text{ then goto } l_1 \text{ else goto } l_2;
\end{align*}
\]

**Input:**

\( In(n) = (2^n, 0) \)

**Output:**

\[
\begin{align*}
\text{Out}(z_1, z_2) &= m, \quad \text{if} \quad (z_1, z_2) = (2^m, 0) \\
\text{Out}(z_1, z_2) &\quad \text{undefined otherwise}
\end{align*}
\]
1. The classical register machines and the BSS model

An example for an infinite dimensional machine

The Model of L. Blum, M. Shub, S. Smale (1989) is similar to:

Registers: \( I_1, I_2, Z_1, Z_2, \ldots \)

Computation and Branching: \( +, -, \cdot, \ldots, \leq \)

Copy: \( Z_1 := Z_j; \ Z_j := Z_1; \)

Input and Output:

\[
(x_1, \ldots, x_n) \in \bigcup_{i \geq 1} \mathbb{R}^i \quad \text{identified} \quad \quad (x_1, \ldots, x_n, 0, 0, \ldots) \in \mathbb{R}^\omega, \ x_n \neq 0
\]

\[
In(x_1, \ldots, x_n, 0, 0, \ldots) = (1, 1, x_1, n, x_2, 0, \ldots, x_n, 0, 0, \ldots), \ x_n \neq 0
\]

\[
Out(i_1, i_2, z_1, z_2, \ldots) = (z_1, z_3, z_5, \ldots) \in \mathbb{R}^\omega
\]

K. Meer (1992): \( In(x_1, \ldots, x_n, 0, 0, \ldots) = (1, 1, x_1, 0, x_2, 0, \ldots, x_n, 0, 0, \ldots) \)

E. Grädel (2007): \( In(x_1, \ldots, x_n, 0, 0, \ldots) = (1, 1, x_1, x_2, \ldots, x_n, 0, 0, \ldots) \)

---

gassnerc@uni-greifswald.de
1. The classical register machines and the BSS model

**An further example**

**for an infinite dimensional machine**

By C. Gaßner (1996), ....

Registers: \( I_1, I_2, \ldots, I_k, Z_1, Z_2, \ldots \)

Computation and branching: \( +, -, \cdot, \ldots, r_j(Z_{k1}, \ldots, Z_{knj}) \) Copy: \( Z_{ik} := Z_{ij} \)

Input (cp. P. Koiran, 1994) and output space:

\[
(x_1, \ldots, x_n) \in \bigcup_{i \geq 1} \mathbb{R}^i
\]

Input:

\[
In(x_1, \ldots, x_n) = (n, 1, \ldots, 1, x_1, \ldots, x_n, 0, 0, \ldots)
\]

Output:

\[
Out(i_1, i_2, \ldots, i_k, z_1, z_2, \ldots) = (z_1, \ldots, z_{i_1}) \in \bigcup_{i \geq 1} \mathbb{R}^i
\]

gassnerc@uni-greifswald.de
1. The classical register machines and the BSS model

The general BSS machine and the input

The **input**: \((Z_1, \ldots, Z_n) := (x_1, \ldots, x_n); I_1 := n; I_2 := 1; \ldots; I_{k_M} := 1;\) 

\[\in \bigcup_{i \geq 1} \mathbb{R}^i\]

![Diagram of the BSS machine and the input](image)
The general BSS machine and the input

The input: \((Z_1, \ldots, Z_n) := (x_1, \ldots, x_n); \ I_1 := n; \ I_2 := 1; \ldots; I_{k\lambda} := 1;\)
1. The classical register machines and the BSS model

The general BSS machine and the input

The input: \((Z_1, \ldots, Z_n) := (x_1, \ldots, x_n); \ I_1 := n; \ I_2 := 1; \ldots; \ I_{kM} := 1;\)
1. The classical register machines and the BSS model

The non-deterministic BSS machines
(input and guessing)

The guessing:

\[(Z_{n+1}, \ldots, Z_{n+m}) := (y_1, \ldots, y_m) \in \bigcup_{i \geq 1} U^i\]

non-deterministic!

Arbitrary elements can be guessed!

\[x_1 \quad x_2 \quad x_3 \quad x_4 \]
\[Z_1 \quad Z_2 \quad Z_3 \quad Z_4 \quad Z_5 \quad Z_6 \quad Z_7 \quad Z_8 \quad Z_9 \quad Z_{10} \quad Z_{11} \quad \ldots\]

\[y_1 \quad y_2 \quad y_3 \quad y_4 \quad y_5 \quad \ldots\]

Arbitrary elements can be guessed!

gassnerc@uni-greifswald.de
1. The classical register machines and the BSS model

The general BSS machine and the output

The output: \((Z_1, \ldots, Z_{I_1})\)

The size of the output

\[\begin{array}{cccccccccccc}
Z_1 & Z_2 & Z_3 & Z_4 & Z_5 & Z_6 & Z_7 & Z_8 & Z_9 & Z_{10} & Z_{11} & \ldots \\
3 & 7 & 1 & 6 & 1 & 2 & 2 & 1 & 8 & & & \\
\end{array}\]
BSS machine for the problem of ordered tuples

Input: a tuple of integers $(x_1, \ldots, x_n) \in \bigcup_{i \geq 1} \mathbb{N}^i$

Problem: Is the input ordered?

Decision:

```
i := 1;
1: if i = n
   then $x_1 := 1$;
   else
       if $x_i \leq x_{i+1}$
          then $i := i + 1$;
          goto 1;
       else $x_1 := 0$;
```

Output: $x_1$
BSS machine for the computation of the sum

Input: a tuple of integers $(x_1, ..., x_n) \in \bigcup_{i \geq 1} \mathbb{N}^i$
Output: $\sum_{i=1}^{n} x_i$

Computation:

\[
i := 1; \\
i:\text{ if } i < n \text{ then } i := i + 1; \\
x_1 := x_1 + x_i; \\
goto 1; \\
\]

for $i := 2$ to $n$ do
\{
\}
\{ \\
x_1 := x_1 + x_i; \\
\}

Input: $(x_1, ..., x_n)$

\[
i := 1 \\
i < n? \\
no \rightarrow i := i + 1 \\
yes \rightarrow x_1 := x_1 + x_i \\
\]

Output: $x_1$

gassnerc@uni-greifswald.de

1. The classical register machines and the BSS model
1. The classical register machines and the BSS model

The simulation of machines by BSS machines

- A finite dimensional register machine
  - A classical RAM
  - A Type-2 machine with \( I, O \subseteq \{0, 1\}^* \)
  - A BSS machine
- A system (family) of RAM’s for k-dimensional inputs \((k \in \mathbb{N}^+)\)
- An online RAM with \( I, O \subseteq \mathbb{R}^\omega \)
  - A Type-2 machine with \( I, O \subseteq \{0, 1\}^\omega \)
  - A RAM with one-way write-only output tape

\( \Rightarrow \) means „can be simulated by“

Gassnerc@uni-greifswald.de
## 2. Comparison of BSS model and RAM’s

Our BSS model and classical RAM’s with \( I \subseteq \bigcup_{i \geq 1} \mathbb{R}^i \)

<table>
<thead>
<tr>
<th></th>
<th>BSS machine</th>
<th>Classical RAM’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>In-, output mode</td>
<td>offline</td>
<td>online</td>
</tr>
<tr>
<td>In-, output space</td>
<td>( I, O = \bigcup_{i \geq 1} \mathbb{R}^i )</td>
<td>( I \subseteq \mathbb{N}^n ), ( O \subseteq \mathbb{N} )</td>
</tr>
<tr>
<td>Indirect addressing</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Any machine has its own program.</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Number of registers</td>
<td>( \infty )</td>
<td>( &lt; \infty )</td>
</tr>
<tr>
<td>Number of ‘tapes’</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Input</td>
<td>input procedure</td>
<td>read instruction</td>
</tr>
<tr>
<td>Output</td>
<td>output procedure</td>
<td>print instruction</td>
</tr>
</tbody>
</table>
### 2. Comparison of BSS model and RAM’s

#### Our BSS model and classical RAM’s with $I \subseteq \bigcup_{i \geq 1} \mathbb{R}^i$ (Offline)

The end of the input is not decidable by a classical RAM with READ since there is not a blank symbol.

<table>
<thead>
<tr>
<th></th>
<th>BSS model</th>
<th>Classical RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>The size of each input</td>
<td>is stored in an index register.</td>
<td>cannot be computed.</td>
</tr>
<tr>
<td>The output of the last value of each input</td>
<td>is possible.</td>
<td>is not possible.</td>
</tr>
<tr>
<td>The sum of all the input values</td>
<td>can be computed.</td>
<td>cannot be computed.</td>
</tr>
</tbody>
</table>
2. Comparison of BSS model and RAM’s

The classical real RAM without input and output procedure

Inputs are read by a READ instruction

Processor registers and registers of the storage

Outputs are written by a PRINT instruction

---

gassnerc@uni-greifswald.de
2. Comparison of BSS model and RAM’s

The classical real RAM allows the online mode. Therefore, we want to compare the BSS machines, the RAM’s, and the Type-2 machines.

Inputs are read by a READ instruction

```
3.4  π  0.1  0  19  45  1.2  e  -1  3  89  ...
```

Outputs are written by a PRINT instruction
The Type-2 machines
with $Y_1, Y_0 = \{0, 1\}^*$ or $Y_1, Y_0 = \{0, 1\}^\omega$

An one-way read-only input tape

A work tape

A work tape

An one-way write-only output tape

3. Comparison of Type-2 machines, BSS model, and the classical RAM’s
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \(\{0, 1\}\) to a \(\{0, 1\}\)-BSS machine

An one-way read-only input tape

\[
\begin{array}{ccccccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B & B & B & B \\
\end{array}
\]

A work tape

\[
\begin{array}{ccccccccccc}
\ldots & B & B & B & 1 & 0 & 1 & 1 & B & B & B & \ldots \\
\end{array}
\]

Two traces by stretching

\[
\begin{array}{ccccccccccc}
\ldots & B & B & B & B & 1 & 1 & B & 0 & B & B & \ldots \\
\end{array}
\]

A work tape

An one-way write-only output tape

\[
\begin{array}{ccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & B & \ldots \\
\end{array}
\]

\[\text{gassnerc@uni-greifswald.de}\]
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B & B & B & B & \ldots
\end{array}
\]

A work tape

\[
\begin{array}{cccccccc}
\ldots & B & B & B & 1 & 0 & 1 & 1 & B & B & B & \ldots
\end{array}
\]

Two traces by stretching

A work tape

\[
\begin{array}{cccccccc}
\ldots & B & B & B & B & 1 & 1 & B & 0 & B & B & \ldots
\end{array}
\]

An one-way write-only output tape

\[
\begin{array}{ccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & B & \ldots
\end{array}
\]

\[\text{gassnerc@uni-greifswald.de}\]
An one-way read-only input tape

A work tape

Two traces by stretching

A work tape

An one-way write-only output tape

3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine
3. **Comparison of Type-2 machines, BSS machines, and classical RAM’s**

**From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine**

- An one-way read-only input tape
  
  \[
  1 \ 1 \ 0 \ 0 \ 0 \ 1 \ B \ B \ B \ B \ B \ \ldots
  \]

- A work tape
  
  \[
  \ldots \ B 
  \]

- Two traces by stretching

- A work tape
  
  \[
  \ldots \ B \ B \ B \ B \ 1 \ 1 \ B \ 0 \ B \ B \ \ldots
  \]

- An one-way write-only output tape
  
  \[
  1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ B \ B \ B \ B \ \ldots
  \]

\[\text{gassnerc@uni-greifswald.de}\]
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B \\
\end{array}

A work tape

\begin{array}{cccccccc}
\ldots & B & 1 & 0 & 1 & 1 & B & \ldots \\
\end{array}

\begin{array}{cccccccc}
\ldots & B & 1 & 1 & B & 0 & B & \ldots \\
\end{array}

An one-way write-only output tape

\begin{array}{cccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & B & \ldots \\
\end{array}
From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

1 1 0 0 0 1 B B B B B ...

A work tape

... B 1 0 1 1 B ...

... B 1 1 B 0 B ...

1 0 0 0 0 0 0 0 1 B B B B ...

An one-way write-only output tape

3. Comparison of Type-2 machines, BSS machines, and classical RAM’s
3. **Comparison of Type-2 machines, BSS machines, and classical RAM’s**

**From a Turing machine over \(\{0, 1\}\) to a \(\{0, 1\}\)-BSS machine**

An one-way read-only input tape

\[
1 \ 1 \ 0 \ 0 \ 0 \ 1 \ B \ B \ B \ B \ B \ \ldots
\]

A work tape

\[
\ldots \ B \ 1 \ 0 \ 1 \ 1 \ 1 \ B \ \ldots
\]

\[
\ldots \ B \ 1 \ 1 \ 1 \ B \ 0 \ B \ \ldots
\]

An one-way write-only output tape

\[
1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ B \ B \ B \ B \ \ldots
\]
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

```
1 1 0 0 0 1 B B B B B ...
```

A work tape

```
... B 1 0 1 1 B ...
```

```
... B 1 1 B 0 B ...
```

```
1 1 0 0 0 1 B B B ...
```

An one-way write-only output tape

```
gassnerc@uni-greifswald.de
```
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \( \{0, 1\} \) to a \( \{0, 1\}\)-BSS machine

An one-way read-only input tape

\[
\begin{align*}
1 & \quad 1 & \quad 0 & \quad 0 & \quad 0 & \quad 1 & \quad B & \quad B & \quad B & \quad B & \quad \ldots
\end{align*}
\]

A work tape

\[
\begin{align*}
\ldots & \quad B & \quad 1 & \quad 0 & \quad 1 & \quad 1 & \quad 1 & \quad B & \quad \ldots
\end{align*}
\]

\[
\begin{align*}
\ldots & \quad B & \quad 1 & \quad 1 & \quad B & \quad 0 & \quad B & \quad \ldots
\end{align*}
\]

An one-way write-only output tape

\[
\begin{align*}
1 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 0 & \quad 1 & \quad B & \quad B & \quad B & \quad \ldots
\end{align*}
\]

gassnerc@uni-greifswald.de
From a Turing machine over \(\{0, 1\}\) to a \(\{0, 1\}\)-BSS machine

3. Comparison of Type-2 machines, BSS machines, and classical RAM’s
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\}
to a \{0, 1\}-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B \\
\end{array}
\]

A work tape

\[
\begin{array}{cccccccc}
... & B & 1 & 0 & 1 & 1 & 1 & B & 1 & 0 & B & ... \\
\end{array}
\]

An one-way write-only output tape

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & B & ... \\
\end{array}
\]

gassnerc@uni-greifswald.de
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B \\
\end{array}
\]

A work tape

\[
\begin{array}{cccccccc}
& & 1 & 1 & 0 & 1 & 1 & 0 \\
B & B & B & B & B & B & B & B \\
\end{array}
\]

An one-way write-only output tape

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & B & B & B & B & B & B & B \\
\end{array}
\]
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

1 1 0 0 0 1 B B B B B ...

A work tape

... B B 1 1 0 1 1 B 1 0 B B ...

An one-way write-only output tape

1 0 0 0 0 0 0 0 0 1 B B B B ...

gassnerc@uni-greifswald.de
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\}
to a \{0, 1\}-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B \\
\end{array}
\ldots
\]

An one-way write-only output tape

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{array}
\ldots
\]
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

| 1 | 1 | 0 | 0 | 0 | 1 | B | B | B | B | ... |

An one-way write-only output tape

| 0 | 1 | 1 | B | 1 | 0 | B | B | ... |

\[ \lambda \]

\[ \lambda \]

\[ \lambda \]
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \( \{0, 1\} \) to a \( \{0, 1\}\)-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B \\
\end{array}
\]

\[ \ldots \]

An one-way write-only output tape

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & B & B & B & B \\
\end{array}
\]

\[ \ldots \]
From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

A right unbounded and a left unbounded work tape

3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

gassnerc@uni-greifswald.de
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \(\{0, 1\}\) to a \(\{0, 1\}\)-BSS machine

A right unbounded and a left unbounded work tape

\[\begin{array}{ccccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B & B & B & \ldots \\
\end{array}\]

\[\begin{array}{ccccccccc}
0 & 1 & 1 & B & 1 & 0 & B & B & \ldots \\
\end{array}\]

\[\begin{array}{ccccccccc}
\ldots & B & B & 1 & 1 \\
\end{array}\]

\[\begin{array}{ccccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & B & \ldots \\
\end{array}\]

gassnerc@uni-greifswald.de
From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

A right unbounded and a left unbounded work tape

gassnerc@uni-greifswald.de
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \( \{0, 1\} \) to a \( \{0, 1\} \)-BSS machine

A right unbounded and a left unbounded work tape

gassnerc@uni-greifswald.de
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

A right unbounded and a left unbounded work tape

\[ \text{M} \]

\[ 1 1 0 0 0 1 B B B B B ... \]

\[ 0 1 1 B 1 0 B B B ... \]

\[ 1 1 B B B ... \]

\[ 1 0 0 0 0 0 0 0 1 B B B B ... \]
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \( \{0, 1\} \) to a \( \{0, 1\}\)-BSS machine

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B \\
B & B & B & B & B & B & \ldots
\end{array}
\]

\[
\begin{array}{cccccccc}
0 & 1 & 1 & B & 1 & 0 & B & B \\
\ldots
\end{array}
\]

\[
\begin{array}{cccccccc}
1 & 1 & B & B & \ldots
\end{array}
\]

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & B & B & \ldots
\end{array}
\]

\[
gassnerc@uni-greifswald.de
\]
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

\begin{align*}
1 & 1 0 0 0 1 B B B B B \ldots \\
0 & 1 1 1 B B 0 B \ldots \\
1 & 1 B B \ldots
\end{align*}

An one-way write-only output tape

\begin{align*}
1 & 0 0 0 0 0 0 0 1 B B B B \ldots
\end{align*}

gasserc@uni-greifswald.de
From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

1  1  0  0  0  1  B  B  B  B  B  ...

An one-way write-only output tape

1  0  0  0  0  0  0  0  1  B  B  B  B  ...

\[ M \]

3. Comparison of Type-2 machines, BSS machines, and classical RAM's
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B & B & B & \ldots \\
\end{array}
\]

An one-way write-only output tape

\[
\begin{array}{cccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & B & \ldots \\
\end{array}
\]

gassnerc@uni-greifswald.de
From a Turing machine over \( \{0, 1\} \)
to a \( \{0, 1\} \)-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B & B & B & \ldots \\
\end{array}
\]

A work tape

\[
\begin{array}{cccccccccc}
1 & 0 & 1 & 1 & B & B & B & 0 & B & \ldots \\
\end{array}
\]

\[
\begin{array}{cccccccccc}
1 & 1 & B & B & B & B & B & \ldots \\
\end{array}
\]

An one-way write-only output tape

\[
\begin{array}{cccccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & \ldots \\
\end{array}
\]

3. Comparison of Type-2 machines, BSS machines, and classical RAM’s
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \( \{0, 1\} \) to a \( \{0, 1\}\)-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B & B & B & B & \ldots \\
\end{array}
\]

A work tape

\[
\begin{array}{cccccccc}
0 & 1 & 1 & 1 & B & B & B & 0 & B & \ldots \\
\end{array}
\]

An one-way write-only output tape

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & B & \ldots \\
\end{array}
\]

\( M \)

\[ \text{gassnerc@uni-greifswald.de} \]
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \(\{0, 1\}\) to a \(\{0, 1\}\)-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B \\
\end{array}
\]

A work tape

\[
\begin{array}{cccccccccccc}
0 & 1 & 1 & 1 & 1 & 0 & B & B & B & B & 0 & B & B \\
\end{array}
\]

An one-way write-only output tape

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}
\]

\[\mathcal{M}\]

gassnerc@uni-greifswald.de
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \{0, 1\} to a \{0, 1\}-BSS machine

An one-way read-only input tape

1 1 0 0 0 1 B B B B B ...

A work tape

0 1 1 1 1 B B B B B 0 B B ...

An one-way write-only output tape

1 0 0 0 0 0 0 0 1 B B B B ...

\(\mathcal{M}\)
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Turing machine over \(\{0, 1\}\) to a \(\{0, 1\}\)-BSS machine

An one-way read-only input tape

\[
\begin{array}{cccccccc}
1 & 1 & 0 & 0 & 0 & 1 & B & B & B & B & B & \ldots
\end{array}
\]

A work tape

\[
\begin{array}{cccccccc}
0 & 1 & 1 & 1 & 1 & B & B & B & B & B & 0 & B & B & \ldots
\end{array}
\]

An one-way write-only output tape

\[
\begin{array}{cccccccc}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & B & B & B & \ldots
\end{array}
\]

\begin{align*}
gassnerc@uni-greifswald.de
\end{align*}
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From an input tape of Type-2 machine to a work tape

An one-way read-only input tape

1 1 0 0 0 1 B B B B B ...

A work tape

0 1 1 1 1 1 B B B B B 0 B B ...

An one-way write-only output tape

1 0 0 0 0 0 0 0 1 B B B B ...

gassnerc@uni-greifswald.de
From an input tape of Type-2 machine to a work tape

An one-way read-only input tape

1 1 0 0 0 1 B B B B B ...

An one-way write-only output tape

1 0 0 0 0 0 0 0 1 B B B B ...

gassnerc@uni-greifswald.de
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From an input tape of Type-2 machine to a work tape

An one-way write-only output tape
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From an input tape of Type-2 machine
to a work tape

An one-way write-only output tape
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From an input tape of Type-2 machine to a work tape
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From an input tape of Type-2 machine to a work tape

A new work tape

1 0 1 1 0 1 0 1 0 1 1 B B B ...

An one-way write-only output tape

1 0 0 0 0 0 0 0 1 B B B ...

gassnerc@uni-greifswald.de
From an input tape of Type-2 machine with $Y_1, Y_0 = \{0, 1\}^*$ to a BSS input

The initial configuration for the ‘Type-2’ machine and for input 110001:

The symbol B after the input from $\{0, 1\}^*$

The output tape

The initial configuration for the BSS input of $(1, 1, 0, 0, 0, 1)$:

An index register contains the length of input
From an output tape of Type-2 machine with $Y_1, Y_0 = \{0, 1\}^*$ to a BSS output

The output from the work space into $\bigcup_{i \geq 1} \{0, 1\}^i$
3. Comparison of Type-2 machines, BSS machines, and classical RAM’s

From a Type-2 machine with $Y_1, Y_0 = \{0, 1\}^\omega$

... to a $\{0, 1\}$-RAM...

gassnerc@uni-greifswald.de
## Consequences

Simulation of offline RAM’s without input tape

<table>
<thead>
<tr>
<th>Input space of the RAM</th>
<th>Simulation</th>
<th>Tuples are represented by</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\bigcup_{i \geq 1} {0, 1}^i$</td>
<td>Type-2 machines</td>
<td>${0, 1}^* {0}^\omega$</td>
</tr>
<tr>
<td>$\bigcup_{i \geq 1} \mathbb{N}^i$</td>
<td>finite dimensional register machines</td>
<td>$\mathbb{N}^k$ for a fixed $k$</td>
</tr>
<tr>
<td>$\bigcup_{i \geq 1} \mathbb{R}^i$</td>
<td>Turing machines</td>
<td>${0, 1}^*$</td>
</tr>
<tr>
<td>$\bigcup_{i \geq 1} \mathbb{R}^i$</td>
<td>BSS machines</td>
<td>$\bigcup_{i \geq 1} \mathbb{R}^i$</td>
</tr>
</tbody>
</table>
### The domain of reals and consequences

- For RAM’s with one-way write-only output tape we get (cp. K. Weihrauch, 2001):
  
  $\Rightarrow$ (FP) Every finite portion of the output is already determined by a finite portion of the input.

- H. Friedman, R. Mansfield: ’Algorithmic Procedure’ (1992), Theorem 17:
  
  $\Rightarrow$ There is no 1-1 computable mapping of $\mathbb{R}^2$ into $\mathbb{R}$.

Real RAM’s are not really suitable for a theory of complexity over the reals.
4. The Halting problems for several types of machines

**Semi-decidability of Halting problems over the reals**

\[ B \subseteq \bigcup_{i \geq 1} \mathbb{R}^i. \quad B \text{ is decidable if the characteristic function is computable}. \]

\[ H_\mathbb{R} = \{(x_1, \ldots, x_n, \text{Code}(M)) \mid (x_1, \ldots, x_n) \in \bigcup_{i \geq 1} \mathbb{R}^i \]

\[ & \quad \& \text{ } M \text{ is a machine over } \mathbb{R} \& M \text{ halts on } x \} \]

\[ H_\mathbb{R}^{\text{spec}} = \{\text{Code}(M) \mid M \text{ is a machine over } \mathbb{R} \& M \text{ halts on Code}(M)\} \]
4. The Halting problems for several types of machines

**Semi-decidability of Halting problems over the reals**

\[ B \subseteq \bigcup_{i \geq 1} \mathbb{R}^i. \quad B \text{ is decidable if the characteristic function is computable.} \]

\[ H_\mathbb{R} = \{(x_1, \ldots, x_n, \text{Code}(M)) \mid (x_1, \ldots, x_n) \in \bigcup_{i \geq 1} \mathbb{R}^i \]
\[ \text{& } M \text{ is a machine over } \mathbb{R} \text{ & } M \text{ halts on } x \} \]

\[ H_\mathbb{R}^{\text{spec}} = \{\text{Code}(M) \mid M \text{ is a machine over } \mathbb{R} \text{ & } M \text{ halts on } \text{Code}(M)\} \]

**Is \( H_\mathbb{R} \) semi-decidable?** Yes, if the codes are suitable.

- For BSS machines: by one machine
- For finite dim. register machines: by a system of machines (cp. D. Scott)
  
  The number of used registers is unary encoded in the codes of the machines.

- For infinite classical real RAM’s: by one machine
4. The Halting problems for several types of machines

Decidability of Halting problems over the reals

\[ B \subseteq \bigcup_{i \geq 1} \mathbb{R}^i. \]  
\[ B \text{ is decidable if the characteristic function is computable.} \]

\[ H_{\mathbb{R}} = \{(x_1, \ldots, x_n, \text{Code}(M)) \mid (x_1, \ldots, x_n) \in \bigcup_{i \geq 1} \mathbb{R}^i \]
\[ \& \ M \text{ is a machine over } \mathbb{R} \& \ M \text{ halts on } x \}\]

\[ H_{\mathbb{R}}^{\text{spec}} = \{\text{Code}(M) \mid M \text{ is a machine over } \mathbb{R} \& \ M \text{ halts on } \text{Code}(M)\}\]

Can the undecidability of \( H_{\mathbb{R}}^{\text{spec}} \) be shown?

For BSS machines: by diagonalization

For finite dimensional register machines: ? (not by the usual proof)

For infinite classical real RAM’s: by diagonalization
4. The Halting problems for several types of machines

**Reducibility of semi-decidable problems over the reals to the Halting problems**

\[ B \subseteq \bigcup_{i \geq 1} \mathbb{R}^i. \quad B \text{ is decidable if the characteristic function is computable.} \]

\[ H^R = \{(x_1,\ldots, x_n, \text{Code}(M)) \mid (x_1,\ldots, x_n) \in \bigcup_{i \geq 1} \mathbb{R}^i \]
\[ \& M \text{ is a machine over } \mathbb{R} \& M \text{ halts on } x\}\}

\[ H^{spec}^R = \{\text{Code}(M) \mid M \text{ is a machine over } \mathbb{R} \& M \text{ halts on } \text{Code}(M)\}\]

**Can any semi-decidable problem be reduced to the corresponding** \(H^R\)?

For BSS machines: \(\text{yes}\)

For finite dimensional register machines: \(\text{by a system of machines}\)

For infinite classical real RAM’s without READ: \(\text{no, e.g. } \{(x_1,\ldots, x_n) \in \mathbb{R}^\infty \mid \exists i \ (x_i \neq 0)\}\)
5. Remarks to non-deterministic machines

Remarks to the non-deterministic machines

- A feasible real RAM
  (V. Brattka, P. Hertling)
- A non-deterministic Turing machine
- A digital non-deterministic BSS machine (guesses: zeros and ones)
- A non-deterministic BSS machine (guesses: real numbers)
- A very abstract model of thought

means „can be simulated by“
A general BSS model over arbitrary structures

Thank you very much!
Christine Gaßner

I also thank my students Paul Grieger, Franz Huwald, and Isabel Desire Schwende.

A finite dimensional register machine

A classical RAM

A Type-2 machine with $I, O \subseteq \{0, 1\}^*$

A BSS machine

A system (family) of RAM’s for k-dimensional inputs ($k \in \mathbb{N}^+$)