Register Minimization in ACRA

Kapil Vaidya & Mohit Vyas

October 1, 2016
All the material in this slide are taken from following paper: *Decision Problems for Additive Regular Functions* by Rajeev Alur and Mukund Raghothaman.
Table of contents

1 ACRA Definition
   • Formal Definition of ACRA

2 Properties of ACRA
   • Assumptions
   • Characterization of Register Complexity

3 Conclusion
So, what is an ACRA?

Consider a model of coffee shop

\[
\begin{align*}
\# / y & := x \\
C / x & := x + 2 \\
y & := y + 1
\end{align*}
\]

\[
\begin{align*}
S / x & := y \\
\neg S & := \# / y := x \\
C / x & := x + 1
\end{align*}
\]
ACRA Definition

Properties of ACRA

Conclusion

Formal Definition of ACRA

Definition of ACRA

**ACRA**

- ACRA is basically an automata with (say) ’n’ registers which can hold integer values.
- Each transition, in an ACRA, may have any assignment of form $u := v + c$ where ’u’ and ’v’ are registers and ’c’ is a constant.
- It also has a set of final state. Hence it maps a word $w \in L^*$ to an integer.
To get domain of function represented by ACRA, remove output labels and compute language accepted by resultant automata.

An important limitation is that register updates are test-free, and cannot examine the register contents.
Assumptions

Assume that given ACRA is *trimmed* off of any unreachable state.

Assume that all registers in ACRA are *live*.

**Liveness**

A register 'r' is 'live' in state $q$ iff $\exists \sigma, c$ and $q' \in F$ so that $f(q, \sigma) = $(value of r at q)$+c$. Here $f(q, \sigma)$ is value we get after following $\sigma$ from $q$ to $q'$.

It can be easily seen that deleting a 'non-live' register won’t affect the function represented by ACRA.
Example for liveness

Here $x$ live in $q_1$ and $q_3$. $z$ is live in $q_2$ and $q_3$. $y$ is live in $q_1$, $q_2$ and $q_3$.

\[
\begin{align*}
  x &:= x + 1 \\
  y &:= y + 2 \\
  z &:= x
\end{align*}
\]

\[
\begin{align*}
  y &:= x \\
  z &:= z + 2
\end{align*}
\]
$k$-separable ACRA

An example of 3-separable ACRA

\begin{align*}
x &:= x + 1 \\
y &:= y + 2 \\
z &:= x
\end{align*}

\begin{align*}
z &:= z + 2 \\
y &:= x \\
z &:= y
\end{align*}

\begin{align*}
x &:= x + 1 \\
y &:= y + 1 \\
z &:= y
\end{align*}

\begin{align*}
z &:= z + 2 \\
y &:= x \\
z &:= y
\end{align*}
$k$–separable ACRA

An example of non-3-separable ACRA

\[
\begin{align*}
x &:= x + 1 \\
y &:= y + 2 \\
z &:= x
\end{align*}
\]

\[
\begin{align*}
q_1 & \quad \xrightarrow{x} \quad q_2 \\
q_2 & \quad \xrightarrow{z := z + 2} \quad q_2 \\
q_3 & \quad \xrightarrow{y := x} \quad q_2 \\
q_1 & \quad \xrightarrow{y := y + 1, z := y, x := y} \quad q_3
\end{align*}
\]
$k$—separable ACRA

**Definition.**

Registers of an ACRA $M$ are $k$—separable if there is some state '$q'$ and set of registers '$U'$ so that:-

1. $|U| = k$, all registers $v \in U$ are live in $q$, and
2. For all $c \in \mathbb{Z}$, there is a string $\sigma$, so that $\delta(q_0, \sigma) = q$ and for all distinct $u, v \in U$, at $q$, $|\text{val}(u, \sigma) - \text{val}(v, \sigma)| \geq c$. 

Kapil Vaidya & Mohit Vyas

Register Minimization in ACRA
Characterization of Register Complexity

Theorem 1.

Let \( f : \Sigma^* \rightarrow \mathbb{Z}_\perp \) be a function defined by ACRA M. Then register complexity of \( f \) is at least \( k \) iff the registers of M are \( k \)-separable.
Proof idea of $\leftarrow$:

Let $M$ be a 3-separable ACRA with 3 registers $r_1, r_2, r_3$. 

$$\begin{align*}
\text{start} & \quad \rightarrow \\
q_0 & \quad \rightarrow \\
q & \quad \rightarrow \\
q_{r_1} & \quad \rightarrow \\
q_{r_2} & \quad \rightarrow \\
q_{r_3} & \quad \rightarrow
\end{align*}$$
Proof idea of $\leftarrow$:

Let $M'$ be an ACRA equivalent to $M$ with 2 registers $a_1, a_2$. 

\[ \text{start} \rightarrow s_0 \xrightarrow{\sigma} s \xrightarrow{\sigma_{a_1}} q_{a_1} \xrightarrow{\sigma_{a_2}} q_{a_2} \]
Proof idea of $\rightarrow$:

Consider this non-3-separable ACRA.

\[ x := x + 1 \quad y := z + 1 \]
\[ y := y + 2 \quad z := z + 1 \]
\[ z := x \quad x := x - 1 \]
Consider this construction

\begin{align*}
    r_1 &:= r_2 \\
    r_2 &:= r_1
\end{align*}

\begin{align*}
    r_1 &:= r_1 + 1 \\
    r_2 &:= r_2 - 1
\end{align*}
Proof idea of $\rightarrow$:

For an ACRA with $k$-registers, if it is not $k$-separable then for each state, there will be a pair of registers which are not far apart from each. So using the above construction we can always simulate it using $k-1$ registers.
Why are we studying this?

- Problem we want to tackle is register minimization of Streaming String Transducers (SST) where registers store string (instead of integers as in case of ACRA) and operation is concatenation instead of addition.
- We saw determinization of Subsequential Transducers.
- We also saw register minimization of ACRA.
- Combining the ideas of papers we have read may prove useful in register minimization of SST.
Conclusion

Thank You!