Guaranteeing Good Memory Bounds for Parallel Programs

F. Warren Burton
IEEE TSE, Oct. 1996

Presented by Sunae Seo
Department of CS, KAIST
May 20, 2000

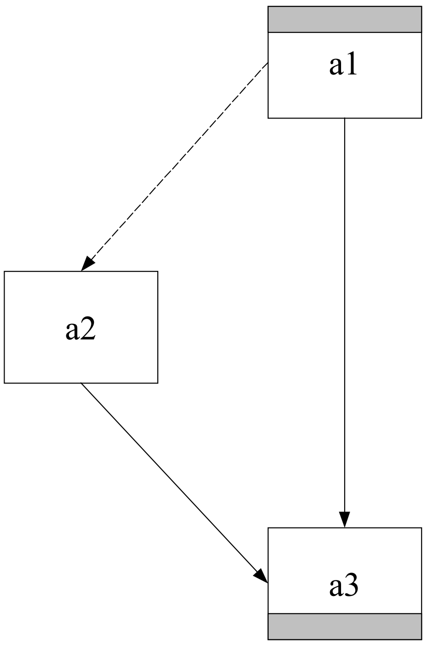
Introduction

- The Goal
  - The program with n units of memory will run in mn units of memory with m processors.
- The Approach
  - Memory amount of a processor at any point $\leq$ Maximal Memory amount of *sequential execution*.
- The Assumptions
  - Idle processor can find work to do within bounded time
  - They don't care the memory fragmentation

Model

- Action
  - unit of program execution
  - possible to de/allocate memory
  - possible to run for a period of time
- **Definition (Time and Memory Task System)** $T = (A, \prec, <, t, \delta)$ where
  - A is a set of actions
  - $\prec$ is a partial order defined on the actions
  - $<$ is total order containing $\prec$
  - t is a timing function(from A to nonnegatice reals)
  - δ is a memory function(from A to integer)

A Simple TAM system



Start Actions & Stint I

Restrictions (A is set of actions)

1. $\Delta(A) = 0$, where $\Delta(A) = \Sigma_{a \in A} \delta(a)$
2. *If* B *is a prefix of* A *w.r.t.* $\prec$*, then* $\Delta(B) \geq 0$

Definitions

- *min, max*
  $min(B)$ is $b \in B$ s.t. $b < c$ for any other $c \in B$
  (max is similar)
- *pred, succ*
  $pred(a)$ is the action immediately before a
  w.r.t. $<$ ($succ$ is similar)
- *Free chain*
  A chain, C is free chain iff
  $\forall c \in C. \forall a \in A. (a \prec c \wedge a \notin C) \supset a \prec min(C)$

Start Actions & Stint II

Definition (Start Action)

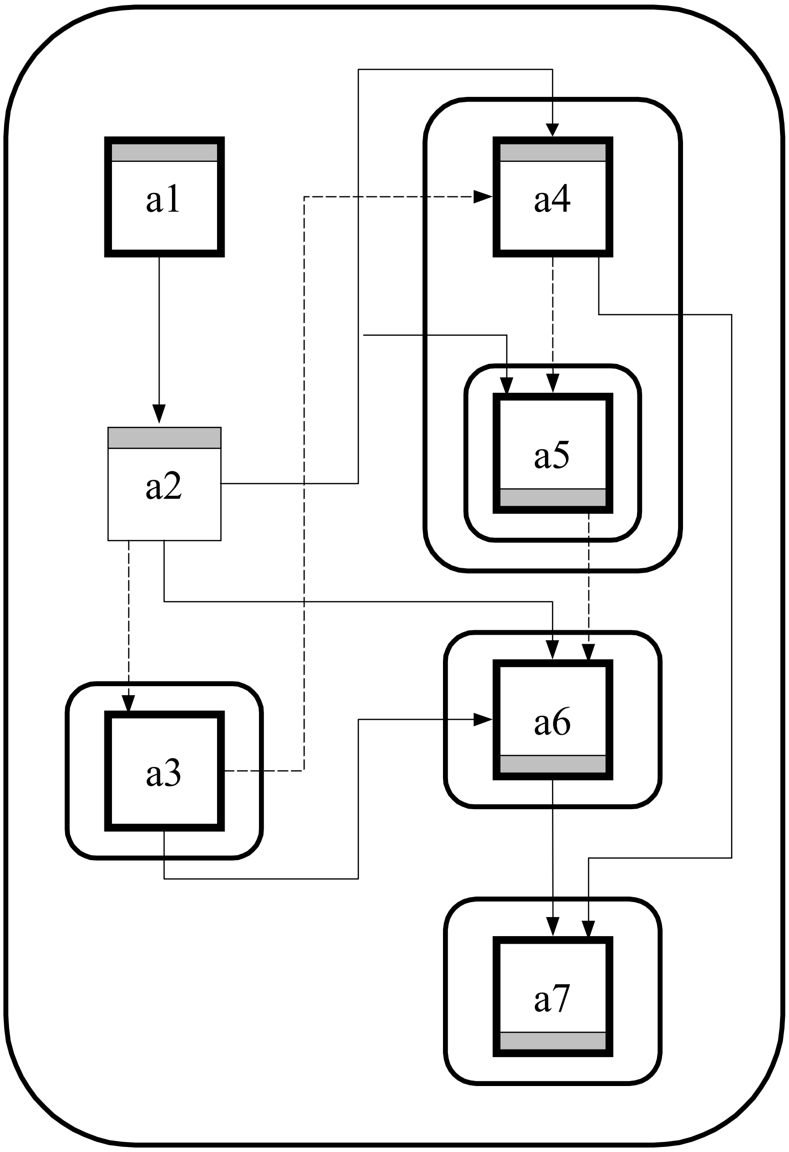
An action, a is called a start action iff
$a = min(A) \vee pred(a) \not\prec a \vee (\exists b \in A.b \prec a \wedge b \not\prec pred(a))$

Definition (Stint)

If a is a start action, then $stint(a)$ is the largest free chain, B, s.t. $a = min(B)$.

- *stint* is a collection of actions executed sequentially as a thread of computation from an *start action*

A Memory-limited TAM system



Memory-limited

Definition (Maximum Sequential Memory Requirement)
The maximum sequential memory requirement of a $chain\ B = \{a_i, a_{i+1}, ..., a_j\}$ is

$$M(B) = max \ \{\Delta(C) | C = a_i, a_{i+1}, ..., a_k, \text{ for } i-1 \leq k \leq j\}$$

Lemma 3. *For any chain, C, contained in the TAM system with actions A, $\Delta(C) \leq M(A)$*

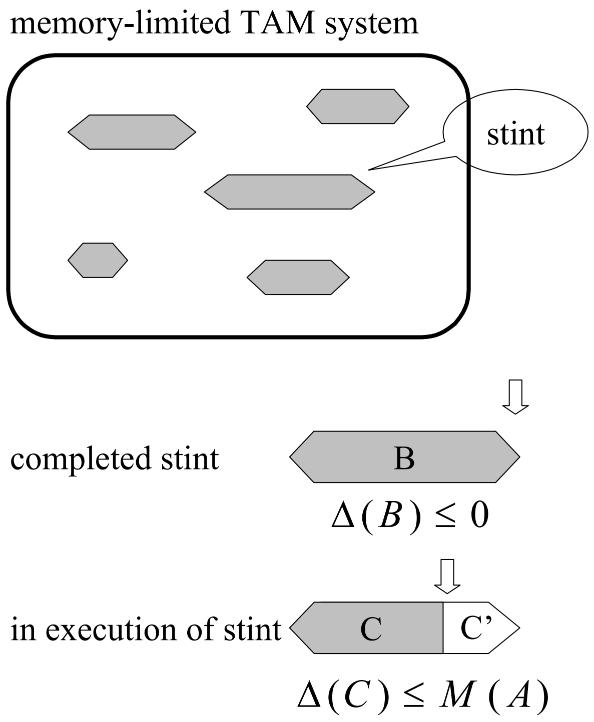
Definition (Memory-limited)

- A free chain, B, is *a memory-limited chain* if $\Delta(B) \leq 0$.
- A TAM system is *a memory-limited TAM system* if every stint in the TAM system is a memory-limited

Theorem 6

If # of processors is m and the maximum sequential memory requirement of memory-limited TAM system is $M(A)$,
then **no more than** $mM(A)$ **units** of memory will be allocated.

$\Rightarrow$ Proof by Def. of memory-limited and Lemma 3



Discussion

- Usability of this method, because action is a unit of execution. - how about recursion?

- TAM system model is too simple and restrictive to be applied to the real programming language.
  - Simple state-transition system
  - Memory-limited Condition

- Some unrealistic assumptions such that parallel programs are deterministic.