Verification of temporal logics on infinite-state systems

Lecture 4.2
Presburger counter systems and acceleration

Stéphane Demri and Valentin Goranko

ESSLLI 2007, Trinity College, August 2007
Overview

Presburger counter systems
  Definition
Linear counter systems
Acceleration
  Definition
Flattable Presburger counter systems
  Definition
  Flattable counter systems

FAST tool
  General presentation
  Specific features
CTL* for admissible counter systems
  Admissible CS
  CTL*
  Problems
  Translation into PA

Procedure
  Flattening
  Completeness
Presburger counter systems

Definition

Linear counter systems

Accelerator

Flattable Presburger counter systems

FAST tool

CTL * for admissible counter systems

Procedure

Linear counter systems

Verification of temporal logics on infinite-state systems

Stéphane Demri and Valentin Goranko

Lecture
Presburger counter systems (PCS) \((\Sigma, Q, T)\)

\[
\begin{align*}
q_0 & \xrightarrow{\psi(x, x')} q_1 \\
q_1 & \xrightarrow{\psi'(x, x')} q_2 \\
q_0 & \xrightarrow{x_1' = x_1 + 1, x_2' = x_2 + 1, x_3' = x_3 + 1} q_1 \\
q_1 & \xrightarrow{x_1' = x_2' = x_3' = 0} q_2
\end{align*}
\]

- Labels: Presburger formulae over current values.
Presburger transition systems (PTS)

Presburger CS
\[ \mathcal{C} = (\Sigma, Q, T) \]
\[ \rightarrow \]
Presburger TS
\[ S_\mathcal{C} = (S, \rightarrow) \]

- \( S = Q \times \mathbb{N}^n \).
- \( \langle q, a \rangle \to \langle q', a' \rangle \) iff \( \exists q \xrightarrow{\psi(x,x')} q' \in T \) s.t. \( a, a' \models \psi(x, x') \).
- Configuration path \( \pi \): infinite path in \( (S, \rightarrow) \).
Linear counter systems

- The definition of Presburger counter systems generalizes the definitions for most usual classes of counter automata.

- Presentation of a Presburger-linear function \((\phi, M, d)\) over \(k\) counters:
  - \(\phi\) is a Presburger formula over \(x_1, \ldots, x_k\) (guard).
  - \(M\) is a \(k \times k\) matrix in \(\mathcal{M}_k(\mathbb{Z})\) over the integers.
  - \(d \in \mathbb{Z}^k\).

- \((\phi, M, d)\) represents the “Presburger formula”:
  \[
  (\vec{x}' = M\vec{x} + d) \land \phi(\vec{x})
  \]

- A linear counter systems is a Presburger counter systems such that each transition is labeled by the presentation of a Presburger-linear function [Finkel & Leroux, FSTTCS 02].
Acceleration
Effect of a loop

\[ \phi(\vec{x}, \vec{x}') \]

◮ How to represent symbolically the set
\[ X = \{ (c, c') \in \mathbb{N}^{2k} : (q, c) (\phi(\vec{x}, \vec{x}'))^* (q, c') \} ? \]

◮ Is \( X \) definable in Presburger arithmetic?

◮ Is \( X \) definable within a symbolic representation adequate for manipulating infinite sets of configurations?
Examples

- \( \phi(\vec{x}, \vec{x}') \equiv x' = x + 1 \Rightarrow x' > x \).

- \( \phi(\vec{x}, \vec{x}') \equiv x' \leq x \Rightarrow x' \leq x \)

- If \( \phi(\vec{x}, \vec{x}') \equiv x' = 2x \) then \( X \) is not definable in Presburger arithmetic (\( \{2^i : i \in \mathbb{N}\} \) is not semilinear).

- Transition table of a Minsky machine can be encoded as a single loop.
  \( \Rightarrow X \) is not definable in Presburger arithmetic.
Linear counter systems with finite monoids

- $\mathcal{M}$: non-empty finite subset of $\mathcal{M}_k(\mathbb{Z})$.
- Multiplicative monoid $\mathcal{M}^*$ generated by $\mathcal{M}$:
  
  $$
  \mathcal{M}^* = \bigcup_{n \geq 0} \bigcup_{M_1, \ldots, M_n \in \mathcal{M}} (M_1 \cdot M_2 \cdots M_n)
  $$

- A linear counter system has the finite monoid property $\iff$ the finite set of matrices labeling the transitions form a finite multiplicative monoid.

  [Boigelot, PhD 98; Finkel & Leroux, FSTTCS 02]

- Examples of such linear counter systems: Minsky machines, Petri nets, reversal-bounded counter automata.

- Finiteness of $\mathcal{M}^*$ is decidable [Mandel & Simon, TCS 77].
Acceleration and finite monoid property

- Given a transition \((q, (\phi, M, d), q)\) such that the monoid \(\{M\}^*\) is finite, one can effectively compute a Presburger formula \(\psi\) with free variables among \(\vec{x}, \vec{x}'\) such that for \(c, c' \in \mathbb{N}^k\), \((q, c)\xrightarrow{\phi, M, d}^* (q, c')\) iff \(c, c' \models \psi\).  

  [Finkel & Leroux, FSTTCS 02]

- **Corollary:** Let \(\mathcal{C}\) be a flat linear counter system with the finite monoid property. For all locations \(q, q'\), one can effectively compute \(\psi\) with free variables among \(\vec{x}, \vec{x}'\) such that for \(c, c' \in \mathbb{N}^k\), \((q, c) \xrightarrow{*} (q', c')\) iff \(c, c' \models \psi\).
Sketch of the proof [Finkel & Leroux, FSTTCS 02]

- We pose
  - $f(c) = Mc + d$
  - $\phi(f^i(\vec{x}))$ for $\exists \vec{y} \phi(\vec{y}) \land \vec{y} = f^i(\vec{x})$ ($i$ is fixed).

- $\{M\}^*$ finite $\Rightarrow$ there are $a, b$ such that $M^{a+b} = M^a$.

- $f^n(c) = M^n c + M^{n-1} d + \cdots + d$.

- $f^n(0) = M^{n-1} d + \cdots + d$.

- $f^{a+b}(c) = M^{a+b} c + M^{a+b-1} d + \cdots + d$
  $= M^{a+b} c + M^{a}(M^{b-1} d + \cdots + d) + (M^{a-1} d + \cdots + d)$
  $= f^a(c) + M^a f^b(0)$
Other calculations

Let us show that $f^{a+qb}(c) = f^a(c) + qM^af^b(0)$.

The proof is by induction on $q$.

- Base case $q = 1$ is immediate.
- By induction hypothesis, $f^{a+(q+1)b}(c) = f^a(f^b(c)) + qM^af^b(0)$.

Using the base case, $f^{a+(q+1)b}(c) = f^a(c) + M^af^b(0) + qM^af^b(0)$.

The formula stating that $(q, c) \xrightarrow{\leq a \text{ steps}} (q', c')$ is below:

$$\bigvee_{0 \leq j < a} (\vec{x}' = f^j(\vec{x}) \land \bigwedge_{0 \leq k < j} \phi(f^k(\vec{x}))).$$
Formula for at least a steps

- Number of steps \( j \geq a \) such that \( j - a = r + qb \) with \( r \in \{0, \ldots, b - 1\} \)

\[
\bigvee_{0 \leq r < b} \exists q \geq 0 \; \psi_1^{r,b} \land \psi' \land \psi_2^{r,b}
\]

\[
\psi_1^{r,b} \equiv \vec{x}' = f^r(f^a(\vec{x}) + qM^a f^b(0))
\]

\[
\psi' = \bigwedge_{0 \leq k < a} \phi(f^k(\vec{x}))
\]

\[
\psi_2^{r,b} \equiv \forall 0 \leq k < r + qb, \; \exists 0 \leq r' < b - 1, \; q' \wedge (k = r' + q'b) \land \phi(f^{r'}(f^a(\vec{x}) + q'M^a f^b(0)))
\]
Flattable Presburger counter systems
Transition sequences

- For $t = (q, \phi, q')$,
  $$\mathcal{R}(t) = \{((q, c), (q', c')) \in (Q \times \mathbb{N}^k)^2 : c, c' \models \phi\}.$$

- $\mathcal{R}(t_1 \cdots t_n) = \mathcal{R}(t_1) \circ \cdots \circ \mathcal{R}(t_n)$.

- $\mathcal{R}((t_1 \cdots t_n)^*) = \mathcal{R}(t_1 \cdots t_n)^*$.

- For any language $L$ whose alphabet is a set of transitions,
  $$\mathcal{R}(L) = \bigcup_{w \in L} \mathcal{R}(w).$$

- For $I \subseteq Q \times \mathbb{N}^k$, $post(L, I)$ is the restriction of $\mathcal{R}(L)$ to pairs with first argument an element of $I$. 
Flattable Presburger counter systems

- Linear path schema: \( \sigma_0 \theta_1^* \sigma_1 \theta_2^* \sigma_2 \cdots \sigma_k \) (sequences \( \sigma_i, \theta_i \)).
- Semilinear path schema: finite set of linear path schemata.
- For all relational counter automata and semilinear path schemata \( \rho \), \( R(\rho) \) is effectively semilinear.

\[ \text{[Comon & Jurski, CAV 98].} \]

- A PCS is globally flattable iff there is a semilinear path schema \( \rho \) such that
  \[
  R(\rho) = \left\{ \left( (q, c), (q', c') \right) \in (Q \times \mathbb{N}^k)^2 : (q, c) \xrightarrow{} (q', c') \right\}.
  \]

- A PCS is initially flattable wrt \( I \subseteq Q \times \mathbb{N}^k \) iff
  \[
  \text{post}(\rho, I) = \left\{ (q', c') : (q, c) \xrightarrow{} (q', c') \& (q, c) \in I \right\}.
  \]
Flattable counter systems

- Many known semilinear classes of counter automata are flattable [Leroux & Sutre, ATVA 05].
  - Subclasses of Petri nets.
  - Subclasses of counter automata.

- Every reversal-bounded counter automata is globally flat.
  [Leroux & Sutre, ATVA 05]

- For all gainy counter automata $\mathcal{A}$ and $I \subseteq Q \times \mathbb{N}^k$, $\mathcal{A}$ is initially flattable wrt $I$ [Leroux & Sutre, ATVA 05].
Sketch of the proof (I)

- Gainy counter automaton \((Q, \delta), I \subseteq Q \times \mathbb{N}^k\).

- For \(q \in Q\), \(X_q = \{ c \in \mathbb{N}^k : \exists (q', c') \in I, (q', c') \overset{*}{\rightarrow} (q, c) \}\).

- Dickon's Lemma [Dickson, 13]

  Any infinite sequence \(c_1, c_2, \ldots\) from \(\mathbb{N}^k\) has two positions \(i < j\) with \(c_i \leq c_j\).

- Consequently,
  - \(X_q\) has a finite set \(\text{Min}(X_q)\) of minimal elements wrt \(\leq\), say \(\text{card}(\text{Min}(X_q)) = N_q\).
  - \(X_q = \{ c \in \mathbb{N}^k : \exists c' \in \text{Min}(X_q), c' \leq c \}\).
Sketch of the proof (II)

For $q \in Q$ and $c_i \in \text{Min}(X_q)$, let $\rho_q^i$ be the finite sequence of transitions such that there is $(q', c') \in l$ with

$$(q', c') \mathcal{R}(\rho) (q, c^i)$$

where

$$\mathcal{R}(\rho) (q, c^i)$$

is a relation such that for all $q \in Q$ and $c_i \in \text{Min}(X_q)$,

$$\text{post} \left( \bigcup_{q \in Q} \bigcup_{i \in \{1, \ldots, N_q\}} \rho_q^i, l \right)$$

is equal to

$$\left\{ (q', c') : (q, c) \rightarrow (q', c') \land (q, c) \in l \right\}.$$
<table>
<thead>
<tr>
<th>Lectures</th>
<th>Verification of temporal logics on infinite-state systems</th>
<th>Stéphane Demri and Valentin Goranko</th>
</tr>
</thead>
<tbody>
<tr>
<td>General presentation</td>
<td>Specific features</td>
<td>Flattable Presburger counter systems</td>
</tr>
<tr>
<td>Presburger counter systems</td>
<td>Acceleration</td>
<td>FAST tool</td>
</tr>
<tr>
<td>CTL* for admissible counter systems</td>
<td>Procedure</td>
<td></td>
</tr>
</tbody>
</table>

FAST tool
FAST: Fast Acceleration of Symbolic Transition systems

- Tool FAST is designed to verify safety properties on counter systems [Bardin & Leroux & Point, CAV 06].

- Safety properties are expressed as reachability questions.

- Members of the FAST tool project
  - Jérôme Leroux, Gérald Point (LABRI)
  - Alain Finkel (LSV), Sébastien Bardin (CEA)
  - Laure Petrucci (LIPN), . . .

- Theoretical foundations: symbolic model-checking, acceleration, flat acceleration [Bardin et al., LSV TR 07].
Main advantages

- Edge-cutting techniques for acceleration, flattening and reduction.

- Fast is complete with respect to flattable counter systems.

- Numerous experiments are successful with Fast and allow to compare with other competitive tools such as
  - LASH [Wolper & Boigelot, CAV 98]
  - TReX [Annichini & Bouajjani & Sighireanu, CAV 01]
  - ALV, BABYLON, BRAIN
Number Decision Diagrams representation (NDD)

- Semilinear sets of tuples in $\mathbb{N}^k$ are represented by finite-state automata with alphabet $\{0, 1\}^k$.

- Set-theoretical operations correspond to operations on automata.

- Presburger-definable sets of tuples can be canonically represented.

- The post and pre operations for ndd-definable relations are also straightforward.

- By contrast Presburger formulae and semilinear sets lack canonicity.
NDD for $x + y = z$
Software architecture

- Client-server architecture:
  - server: computation engine
  - client: front-end which allows the user to interact with the server through a graphical user interface (GUI)

- **Mona** library provides basis for automata manipulations.

- The client is written in Java.

- More can be found on **FAST** web page

  [http://www.lsv.ens-cachan.fr/fast/](http://www.lsv.ens-cachan.fr/fast/)
CTL* for admissible counter systems
Flatness

A PCS is flat if every control state belongs to at most one cycle with no repeated vertex.
Functionality

- A PCS $C$ is functional iff every formula $\psi(x, x')$ labeling a transition in $C$ defines a partial function.

- It is decidable whether a given PCS is functional.

- The reachability problem is not decidable for all:
  - flat linear PCSs.
  - PCSs (Matrix = Id).

[Cortier, TIA 02]
[Minsky, 67]
Counting iteration - Definitions

- \( R \subseteq \mathbb{N}^n \times \mathbb{N}^n \).

\[ \langle a, i, b \rangle \in R_{CI} \text{ iff } \langle a, b \rangle \in R^i. \]

- \( R \) has Presburger counting iteration (pci) iff \( R_{CI} \) is Presburger-definable.

- A PCS \( C \) has pci iff every cycle relation in the control graph of \( C \) has the pci.
A cycle relation with no Presburger counting acceleration

\[ \psi_1(x, x') \lor \psi_2(x, x') \]

\[ \psi_1(x, x') = x_1 > 0 \land x_1' = x_1 - 1 \land (x_2, x_3, x_4) = (x_2', x_3', x_4'). \]

\[ \psi_2(x, x') = x_1 = 0 \land x_2 > 0 \land x_1' = x_3 \land x_2' = x_2 - 1 \land (x_3, x_4) = (x_3', x_4'). \]
Admissible Presburger CS

Definition
A PCS is admissible if it is flat, functional, and has the pci.

- Reachability relation is Presburger-definable for flat PCS with pci, see e.g. [Finkel & Leroux, FSTTCS 02].

- Flatness and functionality are decidable properties.

- pci is conjectured undecidable, see [Leroux, TR LABRI 06].
An almost admissible PCS $\mathcal{C}$

\[
\begin{align*}
    x'_1 &= x_1 + 1 \\
    x'_2 &= x_2 + 1 \\
    x'_3 &= x_3 + 1
\end{align*}
\]

The PCS $\mathcal{C}$ is functional, has the pci but it is not flat.

Local model-checking on $\mathcal{C}$ with FOLTL(Pr) is $\Sigma^1_1$-hard. (Accessibility relation is Presburger-definable.)
FOCTL*(Pr) formulae

\[ \varphi ::= \psi(t) \mid \neg \varphi \mid \varphi \land \varphi \mid X\varphi \mid \varphi U \varphi \mid A\varphi \mid \exists y \varphi. \]

- **Variables:**
  - \(x_0\): control state.
  - \(x_1, \ldots, x_n\): counters.
  - \(y, z, t, \ldots\): auxiliary variables (parameters).

- \(\psi(t)\): Presburger formula with free variables from tuple \(t\).
Satisfaction relation

\[ \pi, i \models_{\text{env}} \varphi \]

- **\( \pi \)**: infinite configuration path of some transition system \( S_C \).
- **\( i \)**: position along \( \pi \).
- **\( \text{env} \)**: environment \( \text{VAR} \rightarrow \mathbb{N} \).
- **\( \varphi \)**: FOCTL*(Pr) formula.
Main clauses of $\models_{\text{env}}$

- $\pi, i \models_{\text{env}} \psi(t)$ iff $\pi(i), \text{env} \models \psi(t)$ in PA,

- $\pi, i \models X\varphi$ iff $\pi, i + 1 \models \varphi$,

- $\pi, i \models_{\text{env}} \exists y \varphi$ iff there is $m \in \mathbb{N}$ such that $\pi, i \models_{\text{env}[y \leftarrow m]} \varphi$,

- $\pi, i \models \varphi U \varphi'$ iff there is some $j \geq i$ s.t. $\pi, j \models \varphi'$ and for $i \leq k < j$, we have $\pi, k \models \varphi$,

- $\pi, i \models A\varphi$ iff for every infinite configuration path $\pi'$ s.t. $\pi'_{\leq i} = \pi_{\leq i}$ we have $\pi', i \models \varphi$. 
Examples of properties

**Determinism**: The reachability graph is deterministic:

\[ AG \bigwedge_{0 \leq i \leq n} \neg \exists y (EX(x_i = y) \land EX(x_i \neq y)) \].

**Boundedness**: The reachability graph is finite:

\[ \exists y AG \bigwedge_{1 \leq i \leq n} x_i \leq y \].

**Increasing chain**: On some path the first counter strictly increases at every step:

\[ EG \exists y (y = x_1 \land X(x_1 > y)) \].
Problems

- **LOCAL MODEL CHECKING**:
  - **input**: configuration \((q, a)\), formula \(\varphi\).
  - **output**: 1 iff for every path \(\pi\) s.t. \(\pi(0) = (q, a)\), we have \(\pi, 0 \models \varphi\) (noted \(\mathcal{D}, (q, a) \models \varphi\)).

- **VALIDITY CHECKING WITH AN INITIAL CONDITION**:
  - **input**:PCS \(\mathcal{D}\), Presburger formula \(\psi_0(x)\), formula \(\varphi\).
  - **output**: 1 iff for every configuration \((q, a)\) satisfying \(\psi_0(x)\), for every configuration \((q', a')\) reachable from \((q, a)\), we have \(\mathcal{D}, (q', a') \models \varphi\).
Translation into Presburger Arithmetic
Encoding configuration paths in Presburger arithmetic

- Control path: infinite control path in $\mathcal{C}$.

- Path schemas:

$$x' = 2x$$

$$x' = x + 1$$

$$x' = x - 1$$

$$x' = 2x$$

$$x' = x$$
Control path description

- Control path description: path schema + counters for cycles.

For admissible PCS,

- every control path has a unique control path description.

- a configuration path is determined by a control path description + an initial configuration.
Local MC is Presburger definable

- Admissible PCS $\mathcal{C}$ (dimension $n$) with $M > 0$ cycles. For all $\alpha \in \mathbb{N}^{M+1}$, $a \in \mathbb{N}^{n+1}$, $m \in \mathbb{N}$ and $b \in \mathbb{N}^{n+1}$

$$\alpha, a, m, b \models \text{PathConfig}_C(\xi, x, i, y)$$

iff $\alpha$ is a valid control path description and the $m^{th}$ configuration of the configuration path $\langle \alpha, a \rangle$ is $b$.

- Local model-checking for FOCTL*(Pr) is decidable.

- Admissible PCS $\mathcal{C}$ (dimension $n$). For every FOCTL*(Pr) formula $\varphi$, one can compute a formula $\psi(x)$ s.t.

$$\text{for every } (q, a) \in S_C, (q, a) \models \psi(x) \text{ iff } D, (q, a) \models \varphi.$$
Post*-Flattening [Bardin et al, ATVA 05]

Let $C = (\Sigma, Q, T)$ and $C' = (\Sigma, Q', T')$ be PCSs, $f : Q' \rightarrow Q$. $(C', q')$ is a $f$-flattening of $(C, q)$ iff

- $f(q') = q$,
- $C'$ is flat,
- $r \xrightarrow{\psi(x,x')} s \in T'$ implies $f(r) \xrightarrow{\psi(x,x')} f(s) \in T$.

$(C', q')$ is a $f$-post*-flattening of $(C, q)$ wrt $\psi(x)$ iff

- $(C', q')$ is a $f$-flattening of $(C, q)$.
- Preservation of reachability sets:

$$\text{post}^{*}_C(q, \psi(x)) = f(\text{post}^{*}_{C'}(q', \psi(x))).$$

$(C', q')$ $f$-flattening of $(C, q)$ and $C'$ admissible. It is decidable whether $(C', q')$ is a post*-flattening of $(C, q)$ wrt $\psi(x)$. 
A fltable non flat PCS

\[ \psi_1 \overset{\text{def}}{=} x \neq 1 \land \psi \]
\[ \psi_2 \overset{\text{def}}{=} x' = 0 \land \psi' \]
\[ \psi_3 \overset{\text{def}}{=} x \neq 0 \land x' = 1 \land \psi'' \]
Another flattable non flat PCS

Stéphane Demri and Valentin Goranko

Verification of temporal logics on infinite-state systems
Trace-flattening

- $(C', q')$ is a $f$-trace-flattening of $(C, q)$ wrt $\psi(x)$ iff
  - $(C', q')$ is a $f$-flattening of $(C, q)$.
  - Preservation of sets of traces:
    \[
    \operatorname{traces}_C(q, \psi(x)) = f(\operatorname{traces}_{C'}(q', \psi(x))).
    \]

- $(C', q')$ $f$-flattening of $(C, q)$ and $C'$ admissible. It is decidable whether $(C', q')$ is a trace-flattening of $(C, q)$ wrt $\psi(x)$.

- Let $(D', q')$ be a post*-flattening [resp. trace-flattening] of the PCS $(C, q)$ wrt $a$. Then, for every formula $\varphi$ in the strict EF fragment [resp. the LTL fragment],
  \[
  D', (q', a) \models \varphi \iff C, (q, a) \models \varphi.
  \]
Model-checking($C$: funct. $PCS + pci$; $\varphi$: $FOLTL(Pr)$)

procedure model-check($C, (q, a), \varphi$)
  1. $found := false$;

  2. while not $found$ do

     2.1 Choose fairly a flattening ($C', q'$) of ($C, q$);

     2.2 if ($C', q'$) is a trace-flattening of ($C, q$) then $found := true$;

  3. return $C', (q', a) \models \varphi$. 

Stéphane Demri and Valentin Goranko
Verification of temporal logics on infinite-state systems
Lecture
Completeness

Theorem

(I) $\text{model-check}(\mathcal{C}, (q, a), \varphi)$ terminates iff $(\mathcal{C}, q)$ has a trace-flattening wrt to $(q, a)$.

(II) When $\text{model-check}(\mathcal{C}, (q, a), \varphi)$ terminates, it returns whether $\mathcal{C}, (q, a) \models \varphi$ holds true.
Model-checking flitable counter systems

- Flat linear counter systems with the finite monoid property has an effectively semilinear reachability relation. [Finkel & Leroux, FSTTCS 02]

- Many subclasses of counter automata with decidable reachability problem have flitable counter automata. [Leroux & Sutre, ATVA 05]

- FAST tool is complete for flitable counter automata.

- Admissible counter systems are flitable and CTL* model-checking can be encoded in Presburger arithmetic. [Demri et al., ATVA 06]
Plan for tomorrow lecture

- Reachability problems for timed automata
  - Introduction to timed automata
  - Nonemptiness problems and other problems
  - Extensions including alternating timed automata

- Timed temporal logics
  - Timed Propositional Temporal Logic (TPTL)
  - Metric temporal logic (MTL) including recent undecidability results
  - Timed CTL
Incomplete bibliography

- **S. Bardin, J. Leroux, and G. Point.**
  FAST Extended Release.

- **S. Demri, A. Finkel, V. Goranko, and G. van Drimmelen.**
  Towards a model-checker for counter systems.

- **A. Finkel and J. Leroux.**
  How to compose Presburger accelerations: Applications to broadcast protocols.

- **J. Leroux and G. Sutre.**
  Flat counter systems are everywhere!