More on the Size of Higman-Haines Sets: Effective Constructions

by

Markus Holzer

Institut für Informatik
Technische Universität München
Boltzmannstraße 3
D-85748 Garching bei München
Germany

September 2007

Joint work with Hermann Gruber (LMU München)
and Martin Kutrib (Universität Gießen).

Outline

Introduction

Motivation.

- Descriptional Complexity
- Recursive *versus* non-recursive trade-offs
- Semi-decidable properties

History.

- Long and fruitful
- Proof schemes for non-recursive trade-offs
- ...

Here.

- Constructability issues of Higman-Haines sets

Higman's Lemma

Lemma (Higman's Lemma)

If X is any set of words formed from a finite alphabet, it is possible to find a finite subset X_0 of X such that, given a word w in X, it is possible to find w_0 in X_0 such that the letters of w_0 occur in w in their right order, though not necessarily consecutively.

G. Higman
(1917–)

References

Higman, G.
Ordering by divisibility in abstract algebras.
Proc. London Math. Soc. 2 (1952), 326–336.

Haines' Theorem

Theorem

Let $L \subseteq A^*$ be an arbitrary language, then both sets

$$\begin{aligned} \textsc{Down}(L) &= \{\, v \in A^* \mid \exists w \in L \text{ s.t. } v \leq w \,\} \\ \textsc{Up}(L) &= \{\, v \in A^* \mid \exists w \in L \text{ s.t. } w \leq v \,\}, \end{aligned}$$

where $\leq$ denotes the scattered subword relation, are regular.

L. H. Haines

References

Haines, L. H.
On free monoids partially ordered by embedding.
J. Combinatorial Theory 6 (1969), 94–98.

Higman's Lemma Rephrased—The Finite Basis Property

Theorem (Higman)

Let L be an arbitrary language. Then there exist words $w_i \in L$ with $1 \leq i \leq n$, for some natural number n which depends only on L, such that

$$\mathrm{Up}(L) = \bigcup_{1 \leq i \leq n} \mathrm{Up}(\{w_i\}).$$

Finite Basis Property. The words $w_1, w_2, \ldots, w_n$ are called a *basis* of L if and only if all words are *minimal*, where a word $w \in L$ is *minimal* in L if and only if there is no $v \in L$ with $v \leq w$ and $v \neq w$.

Examples

Example

Let $A = \{0, 1\}$. Then

$$\lambda, 0, 1, 00, 01, 10, 11, 001, 011, 100, 101, 111, 0011, 1011, 1001, 10011 \leq 10011$$

and $10011 \leq 10011, 010011, 100011, 100101, 100110, \ldots$

Let $L' = (01)^*10$ over the alphabet A. Then

$$\begin{aligned} \textsc{Down}(L') &= ((0+\lambda)(1+\lambda))^*(1+\lambda)(0+\lambda) \\ &= (0+1)^* \quad \text{because} \quad w \leq (01)^{|w|}10 \\ \textsc{Up}(L') &= (A^*0A^*1A^*)^*A^*1A^*0A^* = 0^*1^+0^+(0+1)^*. \end{aligned}$$

Some Easy Properties

Lemma

Let $L \subseteq A^*$ be an arbitrary language, then the following statements hold:

1. *Language L is empty if and only if $\mathrm{Down}(L)$ is empty.*
2. *Language L is finite if and only if the set $\mathrm{Down}(L)$ is finite.*
3. *Language L is empty if and only if $\mathrm{Up}(L)$ is empty.*
4. *Language L contains the empty word λ iff $\mathrm{Up}(L) = A^*$.*

Comment. Higman-Haines sets for languages accepted by Turing machines cannot be effectively constructed (Π_2-completeness in case of down-set problem and Δ_2-completeness w.r.t. Turing reductions for the up-set problem)

Applications

Ehrenfeucht, A., Haussler, D., and Rozenberg, G.
On regularity of context-free languages.
Theoret. Comput. Sci. 27 (1983), 311–332.

Fernau, H. and Stephan, F.
Characterizations of recursively enumerable sets by programmed grammars with unconditional transfer.
J. Autom., Lang. Comb. 4 (1999), 117–152.

Gilmore, R. H.
A shrinking lemma for indexed languages.
Theoret. Comput. Sci. 163 (1996), 277–281.

Applications

Ilie, L.
Decision problems on orders of words.
Ph.D. thesis, Department of Mathematics, University of Turku, Finland, 1998.

van Leeuwen, J.
A regularity condition for parallel rewriting systems.
SIACT News 8 (1976), 24–27.

van Leeuwen, J.
Effective constructions in well-partially-ordered free monoids.
Discrete Mathematics 21 (1978), 237–252.

Is it Effectively Constructible or Not?

Theorem

Let D be a family of automata or grammars.

1. *If for all $M \in D$ a finite automaton accepting $\mathrm{Down}(L(M))$ can effectively be constructed, then there is a recursive function $f : \mathbb{N} \to \mathbb{N}$ such that size $f(|M|)$ is sufficient for a finite automaton to accept $\mathrm{Down}(L(M))$. The statement holds for the up-set as well.*
2. *If there exists a recursive function $f : \mathbb{N} \to \mathbb{N}$ such that for all $M \in D$ size $f(|M|)$ is sufficient for a finite automaton to accept*

$\mathrm{Down}(L(M))$,	$\mathrm{Up}(L(M))$,
then infiniteness is semi-decidable for D.	*then emptiness is semi-decidable for D.*

Well-Known Language Families

Theorem

Let D be a family of automata or grammars which represents the

1. *regular, linear context-free, or context-free languages, then given $M \in D$ there is an effective procedure to construct a finite automaton that accepts $\textsc{Down}(L(M))$.*
2. *recursively enumerable, recursive, context-sensitive, growing context-sensitive, or Church-Rosser languages, then given $M \in D$ there is no effective procedure to construct a finite automagon that accepts $\textsc{Down}(L(M))$.*

The statements hold for the up-set as well.

Proof. Combine previous theorems and consider infiniteness and emptiness problem for the language families. □

Summary of Results

Down-Set.

	Lower bound	Upper bound
NFA	n	n
DFA	$2^{\Omega(\sqrt{n}\log n)}$	2^n
LIN	$2^{\Omega(n)}$	$O\left(\sqrt{2^{n^2+\frac{(3n+6)}{2}\log n-(4+\log e)n}}\right)$
CFL	$2^{\Omega(n)}$	$O(n2^{\sqrt{2^n}\log n})$

Up-Set.

	Lower bound	Upper bound
NFA	n	n
DFA	$2^{\Omega(\sqrt{n}\log n)}$	2^n
LIN	$2^{\Omega(n)}$	$O(\sqrt{2^{(n+2)\log n}})$
CFL	$2^{\Omega(n)}$	$O(\sqrt{n2^{2^n}\log n})$

Comment. Results refer to NFA-acceptance except for DFA entries.

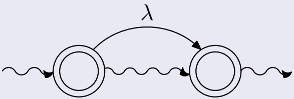
Regular Languages—Finite Automata

Problem. Given a finite automaton M. Determine automaton M' such that it accepts $\text{Down}(L(M))$ ($\text{Up}(L(M))$, resp.).

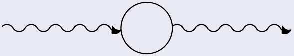
Constructions (Optimizations are Possible).

1. Down-set:

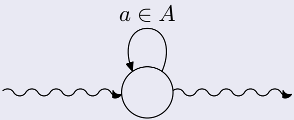
replaced by



2. Up-set:



replaced by



Measure (Size). Number of states of a finite automaton.

Regular Languages—Nondeterministic Finite Automata

Theorem

Let M be a nondeterministic finite automaton of size n. Then size n is sufficient and necessary in the worst case for a nondeterministic finite automaton M' to accept $\textsc{Down}(L(M))$. The finite automaton M' can be effectively constructed.

The statement remains valid for the up-set as well.

Proof. Upper bounds are immediate by construction. Lower bound for down- and up-sets follow from the language $L_n = \{a^{n-1}\}$.

Observe, that the longest word in $\textsc{Down}(L_n)$ and the shortest shortest word in $\textsc{Up}(L_n)$ is of length $n-1$. □

Regular Languages—Deterministic Finite Automata

Theorem

1. *Let M be a deterministic finite automaton of size n. Then size 2^n is sufficient for a deterministic finite automaton M' to accept $\textsc{Down}(L(M))$. The finite automaton M' can be effectively constructed.*
2. *For every n, there exists a language L_n over and $n+2$ letter alphabet, which is accepted by a deterministic finite automaton of size n^2, such that size $2^{n \log n}$ is necessary for any deterministic finite automaton M' accepting $\textsc{Down}(L_n)$.*

The statements remain valid for the up-set as well.

Proof. Upper bounds follow by powerset construction and the aftermentioned observations.

For the lower bound we argue as follows: Let $A = \{a_1, a_2, \ldots, a_n\}$ and $\#, \$ \notin A$. Consider the languages $L_n \subseteq (A \cup \{\#, \$\})^*$ defined as

$$L_n = \{\, \#^j \$ w \in \#^* \$ A^* \mid i = j \bmod n \text{ and } |w|_{a_{i+1}} \leq n \,\}.$$

Language L_n. For each a_i one needs $n+1$ states. For the $\#$-prefix n states are used. This results in

$$n(n+1) + n + 1$$

states for L_n.

Language $\textsc{Down}(L_n)$. One has to keep track of all a_i's simultaneously (counting up to n). This results in

$$n^n + 2$$

states for $\textsc{Down}(L_n)$. □

Down-Sets of Context-Free Languages

Theorem

1. *Let G be a context-free grammar of size n. Then size $O(n2^{\sqrt{2^n}\log n})$ is sufficient for a nondeterministic finite automaton M' to accept $\mathrm{DOWN}(L(G))$. The finite automaton M' can effectively be constructed.*
2. *For every n, there is a language L_n over a unary alphabet generated by a context-free grammar of size $3n+2$, such that size $2^{\Omega(n)}$ is necessary for any nondeterministic finite automaton M' accepting $\mathrm{DOWN}(L(G))$.*

Sketch of Proof. For the upper bound consider context-free grammar $G = (N, T, P, S)$. Iteratively replace the nonterminals on the right hand-side of G by appropriate down-sets obtaining a sequence of grammars $G_0, G_1, \ldots G_{\lfloor \frac{n}{2} \rfloor}$.

For $A \in N$ set $V_A = (N \setminus \{A\}) \cup T$. Define the extended context-free grammar

$$G_A = (\{A\}, V_A, P_A, A)$$

with $P_A = \{\, A \to M \mid (A \to M) \in P \}$, where M in $(A \to M) \in P$ refers to the finite automaton of the right-hand side of the production. For G_A one obtains a finite automaton M_A for $\textsc{Down}(L(G_A))$ as follows:

Observe, that G_A has only one nonterminal.

Distinguish two cases:

1. The production set given by $L(M)$ is linear, i.e., $L(M) \subseteq V_A^*\{A, \lambda\}V_A^*$, or
2. the production set given by $L(M)$ is nonlinear.

For the two cases we proceed as follows:

1. Language $L(M)$ is linear: Construct

   $$L(M_A) = \textsc{Down}(L(M_P)^* \cdot L(M_T) \cdot L(M_S)^*) = \textsc{Down}(L(G_A)),$$

   where

   $$\begin{aligned} L(M_P) &= \{\, x \in V_A^* \mid xAz \in L(M) \text{ for some } z \in (V_A \cup \{A\})^* \,\} \\ L(M_S) &= \{\, z \in V_A^* \mid xAz \in L(M) \text{ for some } x \in (V_A \cup \{A\})^* \,\} \end{aligned}$$

and

$$L(M_T) = L(M) \cap V_A^*.$$

2. Language $L(M)$ is nonlinear: Similar as above (use of an infix set required).

Finally solve recurrence (number of alphabet transitions)

$$|G_k|_t \leq 4 \cdot (|G_{k-1}|_t)^2,$$

for $1 \leq k < \lfloor \frac{n}{2} \rfloor$, describing the substitution step in the kth iteration to construct G_k from G_{k-1}.

For $H_k = \log|G_k|_t$ one obtains

$$H_k \leq 2 \cdot H_{k-1} + 2,$$

which results in

$$|G_{\lfloor \frac{n}{2} \rfloor}|_t \leq 2^{\sqrt{2^n} \log n},$$

because $|G_0|_t \leq n$ and the final step blows up the solution be a factor of four.

Lower bound follows by the context-free grammar

$$G = (\{A_1, A_2, \ldots, A_{n+1}\}, \{a\}, P, A_1)$$

with the productions

$$A_i \;\rightarrow\; A_{i+1}A_{i+1}, \quad \text{for } 1 \leq i \leq n, \text{ and } \quad A_{n+1} \rightarrow a$$

generating the finite unary language $L_n = \{a^{2^n}\}$. □

Up-Sets of Context-Free Languages

Algorithm 1 Determine Basis B of a language $L(G)$

```
i = 0; B_0 = ∅
repeat
    B_{i+1} = B_i ∪ {w} for the shortest word w in L(G) \ Up(B_i)
    i = i + 1
until (L(G) \ Up(B_i)) ≠ ∅
B = B_i
```

Theorem

Let G be a context-free grammar of size n. Then a nondeterministic finite automaton M' of size $O(\sqrt{n 2^{2^n \log n}})$ is sufficient to accept $\mathrm{Up}(L(G))$. The finite automaton M' can effectively be constructed.

Comment. Lower bound as in the case of the down-set problem.

Up- and Down-Sets of Linear Context-Free Languages

Theorem

1. *Let G be a linear context-free grammar of size n. Then a nondeterministic finite automaton M'*

of size $O\left(\sqrt{2^{n^2+\frac{(3n+6)}{2}\log n-(4+\log e)n}}\right)$ *is sufficient to accept* $\textsc{Down}(L(G))$.	*of size* $O(\sqrt{2^{(n+2)\log n}})$ *is sufficient to accept* $\textsc{Up}(L(G))$.

   The finite automaton M' can effectively be constructed.

2. *For every n, there is a language L_n over a binary alphabet generated by a linear context-free grammar of size $12n-2$, such that size $2^{\Omega(n)}$ is necessary for any nondeterministic finite automaton accepting* $\textsc{Down}(L(G))$ *or* $\textsc{Up}(L(G))$.

Discussion

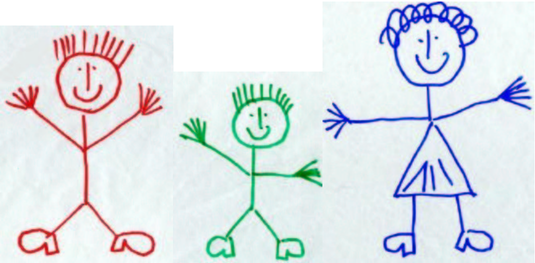
Higman-Haines Sets.

- Continuation of our work on Higman-Haines sets
- Constructability issues of Higman-Haines for:
  - regular languages (det. and nondet. finite automata),
  - linear context-free languages,
  - context-free languages.

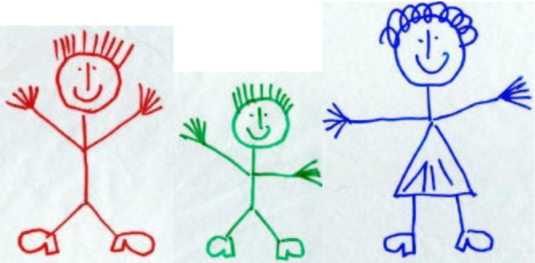
Future work.

- Better bounds for linear context-free and context-free languages
- Other well-quasi orders (Parikh order, etc.)
- . . .

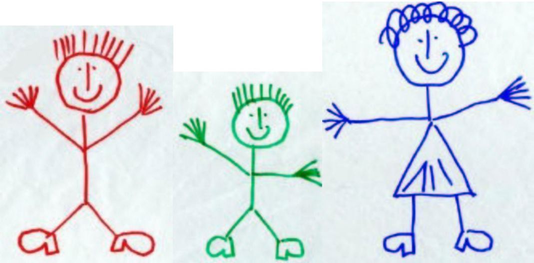
TYFYA



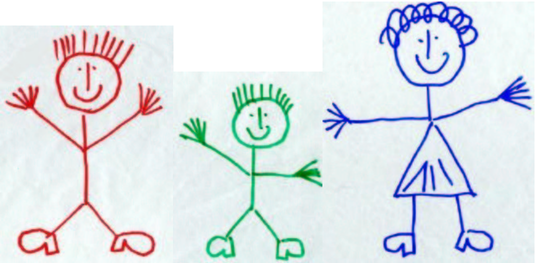
TYFYA



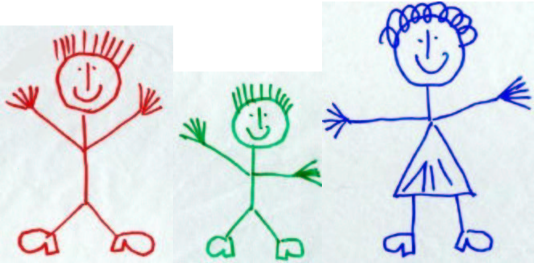
ThYFYA



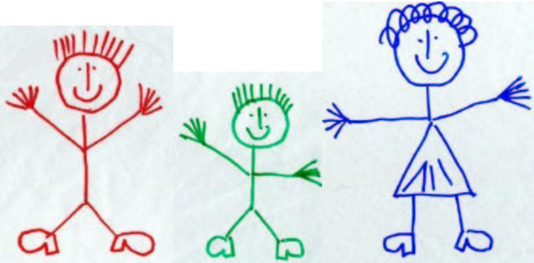
ThaYFYA



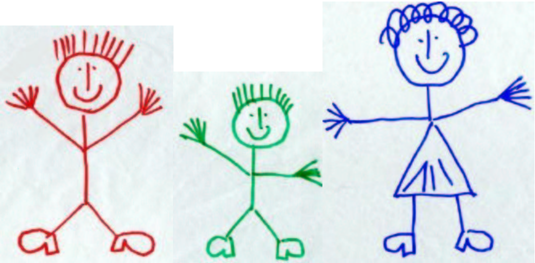
ThanYFYA



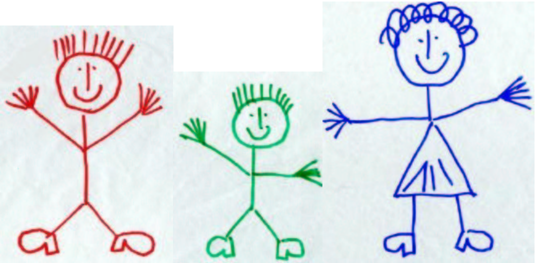
ThankYFYA



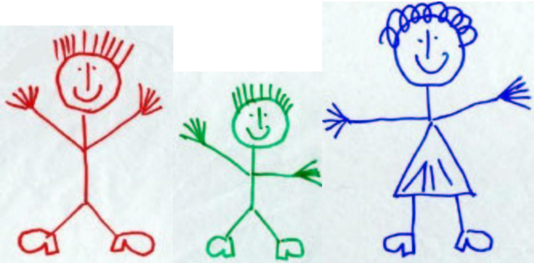
Thank YFYA



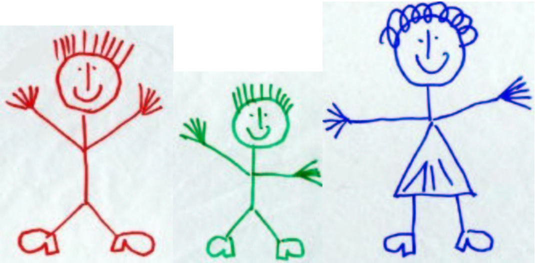
Thank YFYA



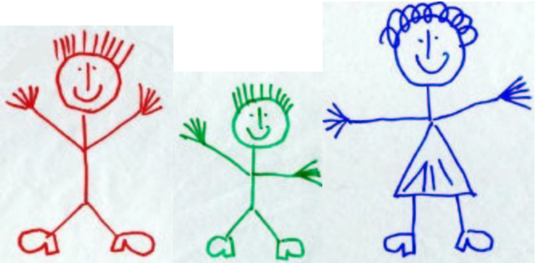
Thank YoFYA



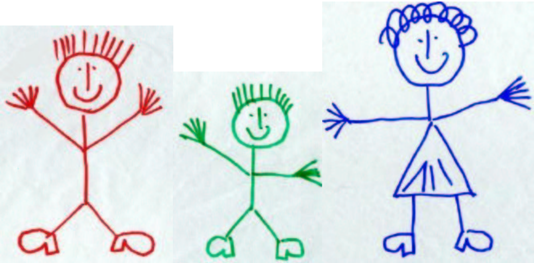
Thank YouFYA



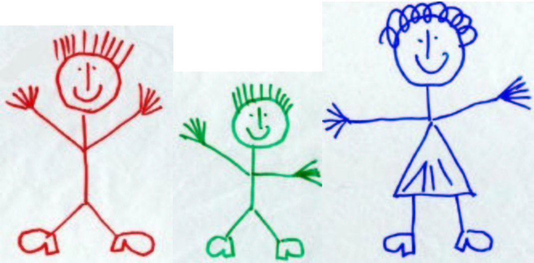
Thank You FYA



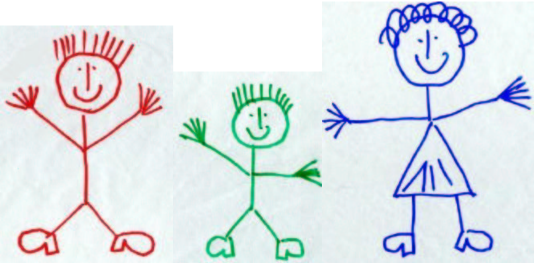
Thank You FYA



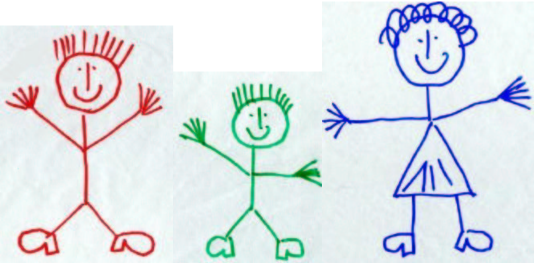
Thank You FoYA



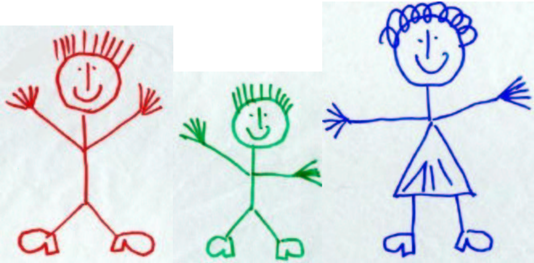
Thank You ForYA



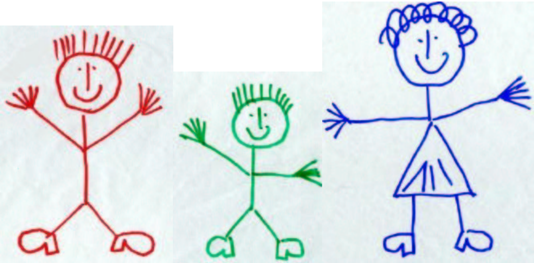
Thank You For YA



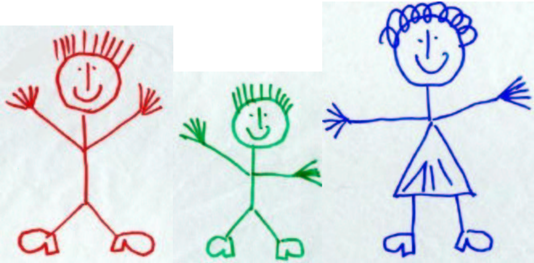
Thank You For YA



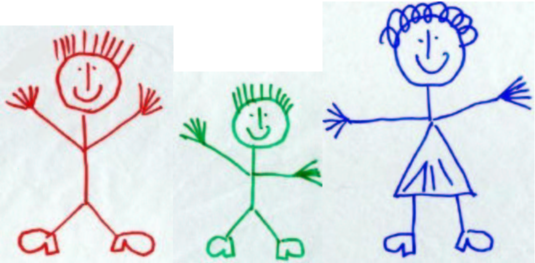
Thank You For YoA



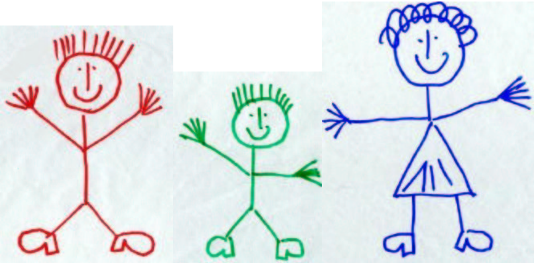
Thank You For YouA



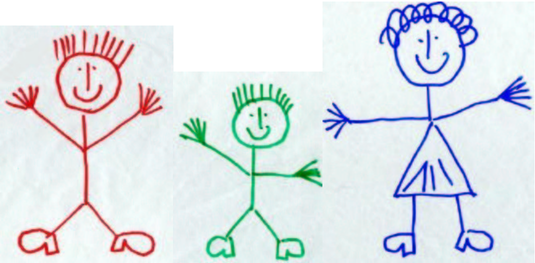
Thank You For YourA



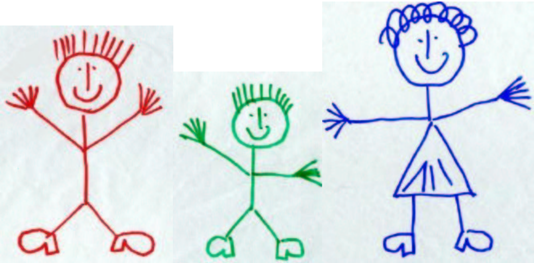
Thank You For Your A



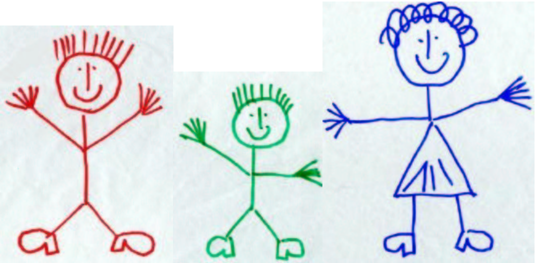
Thank You For Your A



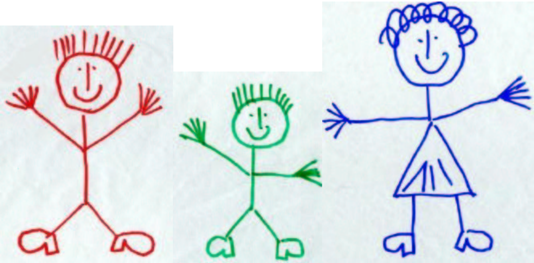
Thank You For Your At



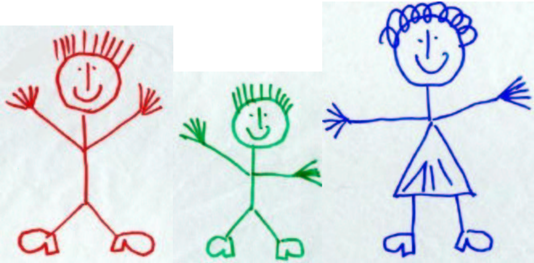
Thank You For Your Att



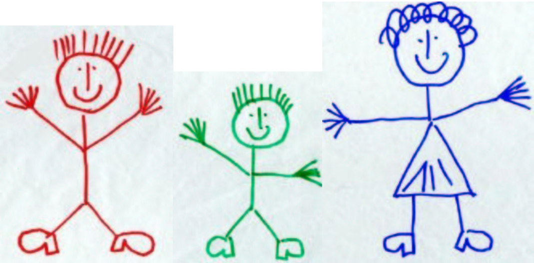
Thank You For Your Atte



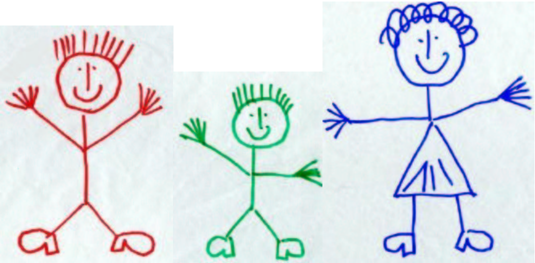
Thank You For Your Atten



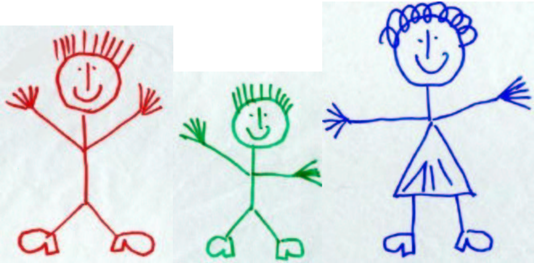
Thank You For Your Attent



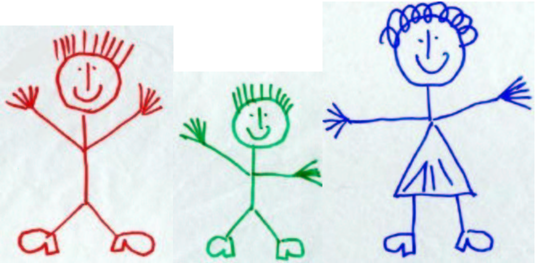
Thank You For Your Attenti



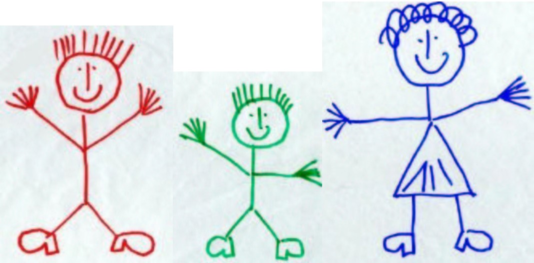
Thank You For Your Attentio



Thank You For Your Attention



Thank You For Your Attention!



Thank You For Your Attention!