Final Exam 2002

May 4, 2004

FINAL EXAM 2002

Each question 1 - 8 carries 25 marks. Answer FOUR of questions 1 - 6.

Question 1. (*25 points*) Let A be the alphabet $\{\texttt{a, c, t}\}$.

(a) Write a Non-Deterministic Finite State Automaton (NFSA) N_A on the alphabet A which accepts texts ending with the keywords `cat` or `ac`.

8 points.

(b) Characterize the language accepted by N_A by giving a regular expression which denotes it.

7 points.

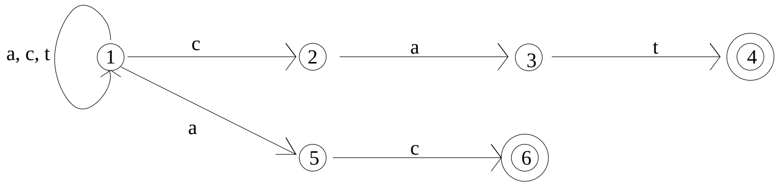
(c) Find a Deterministic Finite State Automaton (DFSA) M_A on the alphabet A which is equivalent to the NFSA N_A considered in part (a) and has the same number of states.

10 points.

Answer: (a) $N_A = \{S, A, \nu, 1, F\}$ where the set of states S is $\{1, 2, 3, 4, 5, 6\}$, the initial state is 1, the final states F are $\{4, 6\}$ and the transition function ν is given by the following table:

	a	c	t
1	$1,5$	$1,2$	1
2	3	$-$	$-$
3	$-$	$-$	4
$*4$	$-$	$-$	$-$
5	$-$	6	$-$
$*46$	$-$	$-$	$-$

The transition diagram of N_A is the following:



(b) The language accepted by N_A is that denoted by the regular expression

$$(\mathbf{a}+\mathbf{c}+\mathbf{t})^*(\mathbf{cat}+\mathbf{ac}).$$

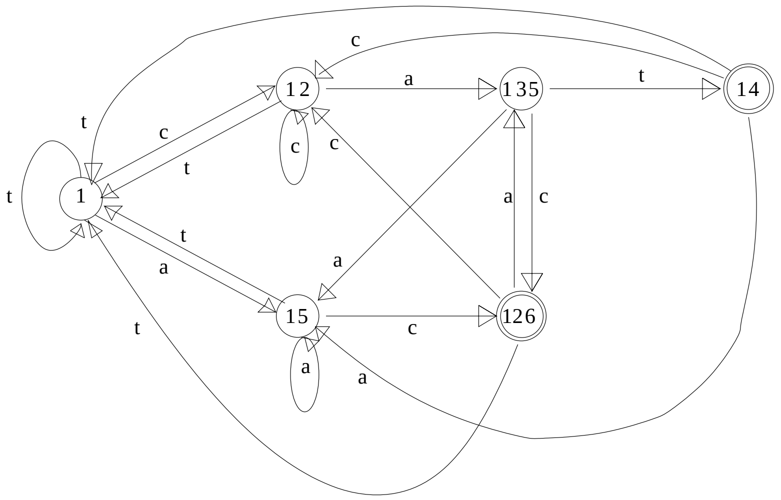
(c) $M_A = \{S, A, \nu', \mathrm{I}, F\}$ where the set of states S is {I, II, III, IV, V, VI} with

$$\mathrm{I} = \{1\} \qquad \mathrm{II} = \{1,2\} \qquad \mathrm{III} = \{1,3,5\}$$

$$\mathrm{IV} = \{1,4\} \quad \mathrm{V} = \{1,5\} \quad \mathrm{VI} = \{1,2,6\}$$

the initial state is I, the final states are IV and VI. The transition function ν' is given by the following table or diagram:

	a	c	t
I	V	II	I
II	III	II	I
III	V	VI	IV
IV	V	II	I
V	V	VI	I
VI	III	II	I



Question 2. (*25 points*) Consider the language $\mathcal{L}$ on the alphabet $A = \{0, 1\}$:

$$\mathcal{L} = \{0^k 1^m 0^{m+k} \mid \text{ for all } \; k, m \in \mathbf{N} \;\}$$

(a) Give a context-free grammar G that generates precisely the language $\mathcal{L}$. (*Hint:* Use three non-terminal symbols S, A and B, where S is the start symbol. To see that G generates precisely $\mathcal{L}$, derive the generic expression).

10 points.

(b) Is $\mathcal{L}$ a regular language? If yes, define an automaton that accepts precisely $\mathcal{L}$; otherwise, use the "pumping lemma" to show that $\mathcal{L}$ is not regular.

15 points.

Answers:

(a) $\mathcal{L}$ is generated by the grammar $G = (\{S, A, B, 0, 1\}, \{0, 1\}, P, S)$ with start symbol S and the following set of production rules P:

$$\begin{array}{lcl} S & \rightarrow & \epsilon \mid A \mid B \\ A & \rightarrow & 0A0 \mid B \mid 00 \\ B & \rightarrow & 1B0 \mid 10 \end{array}$$

To see that G generates $\mathcal{L}$, we derive the generic expression $0^k 1^m 0^{m+k}$:

$$
\begin{aligned}
S &\Rightarrow 0A0 \Rightarrow \ldots (k-1 \text{ times}) \ldots \Rightarrow 0^k A 0^k \Rightarrow 0^k B 0^k \\
&\Rightarrow 0^k 1 B 0 0^k \Rightarrow \ldots (m-2 \text{ times}) \ldots \Rightarrow 0^k 1^{m-1} B 0^{m-1} 0^k \\
&\Rightarrow 0^k 1^{m-1} 1 0 0^{m-1} 0^k = 0^k 1^m 0^{m+k}
\end{aligned}
$$

(b) Let M be an automaton with n states accepting $\mathcal{L}$: consider the expression $0^k 1^m 0^{m+k}$ where $n < k$. Then before reading all of 0^k the machine has been twice in the same state with the same input. It follows that M would also accept a string $0^\ell 1^m 0^{m+k}$ for some $\ell < k$. But $0^\ell 1^m 0^{m+k}$ does not belong to $\mathcal{L}$. (A more detailed argument would be welcome, but the above should suffice.)

Question 3. (*25 points*) (a) Outline an elementary proof of Euclid's theorem: *there are infinitely many prime numbers.*

(10 points)

(b) Implicit in the proof there is an algorithm to define the following function:

$$p(n) = p_n, \qquad \text{the } n\text{-th prime number},$$

starting from $p_1 = 2$. Show that the function $p(n)$ is primitive recursive. You can use the fact that the factorial function $x!$ is primitive recursive and that the relations $x < y$ and $x|y$ (x divides y) are primitive recursive.

(15 points)

(Hint: Consider the predicate $\mathrm{Pr}(x)$ (x *is prime*) defined as

$$\mathrm{Pr}(x) \equiv 1 < x \;\&\; \neg(\exists c. 1 < c < x \;\&\; c|x).$$

First show that $\mathrm{Pr}(x)$ is primitive recursive; then define the function $p(x)$ by the recursion scheme, using the bounded μ-operator.)

Answer: (a) The number 2 is the first prime number, $p_1 = 2$. Let p_n be the n-th prime number and consider $c = p_n! + 1$. The number c is greater than 1 and no prime p_i with $i \leq n$ divides c.
(Indeed, for every $i \leq n$, $p_i | p_n!$, namely, $p_i \cdot e = p_n!$, where $e = p_n \cdot (p_n - 1) \cdot \ldots \cdot (p_i + 1) \cdot (p_i - 1) \cdot \ldots \cdot 1$. If $p_i | p_n! + 1$, i.e., $p_i \cdot d = c$ for some d, then $1 = (p_i \cdot d) - (p_i \cdot e) = p_i \cdot (d - e)$ and this is a contradiction as no prime divides 1.)

Suppose there were only n prime numbers, then c is a prime number and this is a contradiction, since $c > p_n$. Therefore there is a prime number p_{n+1} with $p_n < p_{n+1} \leq c$.

(b) The relation $\mathrm{Pr}(x) \equiv 1 < x \ \& \ \neg(\exists c.1 < c < x \ \& \ c|x)$ is primitive recursive, because it is defined by conjunction, negation and bounded quantification from the primitive recursive relations $x < y$ and $x|y$. Now $p(i)$ is primitive recursive because it is defined by the scheme of primitive recursion using also the composition scheme:

$$\begin{array}{rcl} p(1) & = & 2; \\ p(n+1) & = & \chi(p(n)), \quad \text{where } \chi(c) = \mu x_{c<x\leq c!+1}\mathrm{Pr}(x). \end{array}$$

Question 4. (*25 points*) Let $\mathcal{L} = (M_1, M_2, \ldots)$ be an enumeration of the set of all Turing Machines that compute partial functions from $\mathbf{N}$ to $\mathbf{N}$ and let f_i be the partial function computed by M_i.

(a) What does it mean to say that a Turing Machine M^U is a universal machine for the list $\mathcal{L}$? (You do not need to define one in detail).

(7 points)

(b) Using a universal machine M^U show that there is a Turing Machine M' which computes the following partial function g:

$$\begin{array}{lll} g(n) & = \ 0 & \text{if } M_n \text{ with input } n \text{ returns the value } f_n(n); \\ g(n) & \text{is undefined} & \text{if } M_n \text{ with input } n \text{ does not terminate.} \end{array}$$

(8 points)

(c) Show that there is no Turing Machine M^H which computes the following partial function h:

$$\begin{array}{lll} h(n) & = \ 0 & \text{if } M_n \text{ with input } n \text{ does not terminate;} \\ h(n) & \text{is undefined} & \text{otherwise.} \end{array}$$

(10 points)

Answer. (a) A Turing Machine M^U is universal for the list $\mathcal{L}$ if M^U behaves as an *interpreter* for the Turing machines in $\mathcal{L}$. Namely, M^U takes as input a coding $\underline{M_n}$ of Turing machine M_n (i.e., the program as a datum, representable by the Gödel number of M_n) and an input k and behaves as follows:

$M^U(\underline{M_n}, k)$	terminates with output $f_n(k)$	if $M_n(k)$ terminates with output $f_n(k)$
$M^U(\underline{M_n}, k)$	undefined	if $M_n(k)$ undefined.

(b) Define M' as follows: $M'(\underline{M_n}, n) = 0$ if $M^U(\underline{M_n}, n)$ terminates; $M'(\underline{M_n}, n)$ undefined otherwise.

(c) Suppose M^H computed the partial function h. Then M^H occurs at some point in the list $\mathcal{L}$, say $M^H = M_n$. Now

$h(n)$	$=$	0	if M_n with input n does not terminate in accepting state;
$h(n)$	is undefined		otherwise.

• Suppose $h(n)$ is undefined: by definition of h, M_n with input n terminates in an accepting state. But $M_n = M^H$ computes h, therefore $h(n)$ is defined, a contradiction;
• Suppose $h(n)$ is defined: since M^H computes h, M^H with input n terminates in accepting state; but $M^H = M_n$, so by definition of h, we have $h(n)$ undefined, a contradiction.
In either cases, we have a contradiction; the only remaining assumption, that h is computable by a Turing Machine, is therefore false.

Question 5. (*25 points*) Outline proofs of the theorems in (a) and (b):

(a) The set of all functions $f : \mathbf{N} \to \mathbf{N}$ is not denumerable.

(10 points)

(b) The set of all total unary Turing-computable functions cannot be enumerated by a total Turing-computable function.

(10 points)

(c) Assuming Church's Thesis, can we hope that a better characterization of recursive functions could yield a recursive enumeration of the set of all total recursive functions?

(5 points)

Answers: (a) Given an enumeration f_1, f_2, ..., f_i of all functions $\mathbf{N} \to \mathbf{N}$, define the function

$$g(i) = 3 \quad \text{if} \quad f_i(i) \neq 3,$$
$$g(i) = 2 \quad \text{if} \quad f_i(i) = 3.$$

Then g is a function $\mathbf{N} \to \mathbf{N}$ which is different from all functions in the list; this contradicts the assumption that *all the functions* $\mathbf{N} \to \mathbf{N}$ are in the list.

(b) Suppose $f : \mathbf{N} \to \mathbf{N}$ was a total Turing-computable function such that $f(i)$ is the index of a Turing Machine computing a unary total function. Let $\varphi^U(x, y)$ be the function computed by a Turing Machine M^U which is universal for unary Turing-computable functions. Then the function g defined by

$$g(n) = \varphi^U(f(n), n) + 1$$

is also a unary Turing-computable function, and it is total because each $f(i)$ is the index of a total function; but g is different from all the functions whose index is given by f. Therefore the set of all total Turing-computable functions is not enumerable by a Turing-computable total function.

(c) Church's Thesis claims that every function which is effectively computable by some mechanical procedure (abstraction being made on the resources available for computation) is also computable by a Turing Machine. (This claim is supported by evidence that the various formal definitions so far proposed for effectively computable classes of functions have been proved equivalent to the definition of Turing computability.)
By Church's Thesis, *effectively computable* can be identified with *Turing computable*. Therefore assuming Church's Thesis, (b) shows that the set of all total recursive functions is not recursively enumerable.

Question 6. (*25 points*) Consider the Ackermann function

$$\begin{array}{rcll} \alpha(m, 0) & = & m + 1 & \text{(i)} \\ \alpha(0, n+1) & = & \alpha(1, n) & \text{(ii)} \\ \alpha(m+1, n+1) & = & \alpha(\alpha(m, n+1), n) & \text{(iii)} \end{array}$$

(a) The following Lemma has been proved in class:

Lemma: *For every primitive recursive function* $f(x_1, \ldots, x_k)$ *there exists an* $n \in \mathbf{N}$ *such that*

$$f(x_1, \ldots, x_k) < \alpha(max(x_1, \ldots, x_k), n)$$

for all $x_1, \ldots, x_k$.

The outline of the proof follows in Tables 1 and 2. You must complete the proof by finding numbers $\mathbf{B}_1, \ldots, \mathbf{B}_5$ which satisfy the inequalities in the outline.

(15 points)

(b) Outline the proof given in class that the function $\beta(n) = \alpha(n, n) + 1$ is not primitive recursive. (*Hint:* Use the Lemma.)

(10 points)

PROOF OF THE LEMMA (outline):

The proof is by induction on the definition of a primitive recursive function. There are five cases:

1. *constant functions*: there is a $\mathbf{B}_1$ such that

$$c_0^n(x_1, \ldots, x_n) = 0 < max(x_1, \ldots, x_n) + 1 = \alpha(max(x_1, \ldots, x_n), \mathbf{B}_1).$$

What is $\mathbf{B}_1$ here? (*3 points*)

2. *projection functions*: there is a $\mathbf{B}_2$ such that

$$\pi_i^n(x_1, \ldots, x_n) = x_i < max(x_1, \ldots, x_n) + 1 = \alpha(max(x_1, \ldots, x_n), \mathbf{B}_2).$$

What is $\mathbf{B}_2$ here? (*3 points*)

3. *successor function*: there is a $\mathbf{B}_3$ such that

$$succ(x) = x + 1 < x + 2 = \alpha(x + 1, 0) \leq \alpha(x, \mathbf{B}_3)$$

What is B here? (*3 points*)

CASES 1- 3 (continues in the next table)

Table 1: Proof of the Lemma, cases 1-3

Answer. (a) $\mathbf{B}_1 = 0$; $\mathbf{B}_2 = 0$; $\mathbf{B}_3 = 1$; $\mathbf{B}_4 = max(\mathbf{C}_1, \ldots, \mathbf{C}_k, \mathbf{D})$; $\mathbf{B}_5 = \mathbf{C} + 1$.

(b) Suppose $\beta(n)$ was primitive recursive. Then by the Lemma, there is a k

PROOF OF THE LEMMA (continued):

4. *composition*: let $h(x_1, \ldots, x_m)$ be defined by composition from primitive recursive functions $g(x_1, \ldots, x_k)$ and $f_i(x_1, \ldots, x_m)$ for $i \leq k$. Suppose there is a $\mathbf{D}$ such that for all $y_1, \ldots, y_k$

$$g(y_1, \ldots, y_k) < \alpha(max(y_1, \ldots, y_k), \mathbf{D})$$

and suppose for each $i \leq k$ there is a $\mathbf{C}_i$ such that for all $x_1, \ldots, x_n$,

$$f_i(x_1, \ldots, x_m) < \alpha(max(x_1, \ldots, x_m), \mathbf{C}_i).$$

We show that there is a $\mathbf{B}_4$ such that

$$\begin{array}{rcll}
\alpha(max(x_1, \ldots, x_m), \mathbf{B}_4 + 2) & \geq & \alpha(max(x_1, \ldots, x_m) + 1, \mathbf{B}_4 + 1) & \text{(proved elsewhere)} \\
& = & \alpha(\alpha(max(x_1, \ldots, x_m), \mathbf{B}_4 + 1), \mathbf{B}_4) & \text{(def. of } \alpha) \\
& > & \alpha(max_{i \leq k}\{f_i(x_1, \ldots, x_m)\}, \mathbf{B}_4) & \text{(by hypothesis)} \\
& \geq & \alpha(max_{i \leq k}\{f_i(x_1, \ldots, x_m)\}, \mathbf{D}) & \text{(def. of } \mathbf{B}_4\text{, monot.)} \\
& > & g(f_1(x_1, \ldots, x_m), \ldots, f_k(x_1, \ldots, x_m)). & \text{(by hypothesis)}
\end{array}$$

What is $\mathbf{B}_4$ here? (*3 points*)

5. *recursion*: Let $f(y)$ be a primitive recursive and suppose h is defined by recursion

$$h(0) = 0 \qquad \text{and} \qquad h(n+1) = f(h(n)).$$

Suppose there exists a $\mathbf{C}$ such that $f(y) < \alpha(y, \mathbf{C})$, for all y. We prove by induction that *there exists a $\mathbf{B}_5$ such that $h(x) < \alpha(x, \mathbf{B}_5)$, for all x.*

Base case:

$$h(0) = 0 < 1 = \alpha(0, 0) < \alpha(0, \mathbf{B}_5)$$

Inductive step: Suppose $h(n) < \alpha(n, \mathbf{B}_5)$ (inductive hypothesis).

Then

$$\begin{array}{rcll}
h(n+1) & = & f(h(n)) & \\
& < & \alpha(h(n), \mathbf{C}) & \text{(by assumption)} \\
& < & \alpha(\alpha(n, \mathbf{B}_5), \mathbf{C}) & \text{(by inductive hypothesis)} \\
& = & \alpha(n+1, \mathbf{B}_5) & \text{(by def. of } \mathbf{B}_5 \text{ and of } \alpha)
\end{array}$$

What is $\mathbf{B}_5$ here? (*3 points*)

END OF PROOF OF THE LEMMA.

Table 2: Majorization Lemma

such that $\beta(m) < \alpha(m,k)$ for all m. Therefore

$$\begin{array}{rcll} \alpha(k,k)+1 & = & \beta(k) & \text{definition of } \beta \\ & < & \alpha(k,k) & \text{by the Lemma} \end{array}$$

a contradiction. Therefore $\beta(n)$ is not primitive recursive.

ADDITIONAL QUESTIONS

Question 7. (*25 points*) Let B the alphabet $\{a,b\}$.

(a) Write a Non-Deterministic Finite State Automaton N_B on the alphabet B which accepts precisely the expressions whose penultimate symbol is b.

8 points.

(b) Using the powerset construction, find a Deterministic Finite State Automaton M_B on the alphabet B which is equivalent to the NFSA N_B considered in part (c).

10 points.

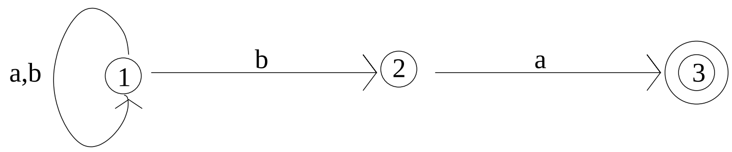
(c) If the DFSA M_B is not minimal, describe a minimal one.

7 points.

Answer. (a) $N_B = \{S, A, \nu, 1, F\}$ where the set of states S is $\{1,2,3\}$, the initial state is 1, 3 is the only final state in F and the transition function ν is given by the following table

	a	b
1	1	1, 2
2	3	3
$*3$	$-$	$-$

The transition diagram of ν is



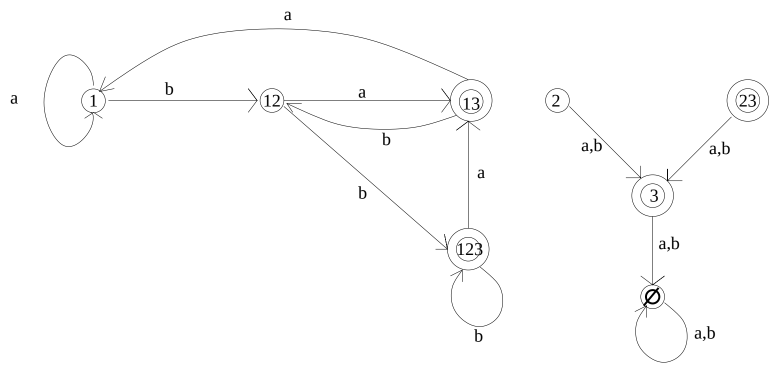
(b) Applying the powerset construction, we obtain:

$$M_B = \{\wp(S), B, \nu', \{1\}, F\}$$

where $\{1\}$ is the initial state and the set F of accepting states is $\{\{3\}, \{1,3\}, \{2,3\}$ and $\{1,2,3\}\}$, and the transition function ν is as follows:

	a	b
$\emptyset$	$\emptyset$	$\emptyset$
$\{1\}$	$\{1\}$	$\{1,2\}$
$\{2\}$	$\{3\}$	$\{3\}$
$\{3\}$	$\emptyset$	$\emptyset$
$\{1,2\}$	$\{1,3\}$	$\{1,2,3\}$
$\{1,3\}$	$\{1\}$	$\{1,2\}$
$\{2,3\}$	$\{3\}$	$\{3\}$
$\{1,2,3\}$	$\{1,3\}$	$\{1,2,3\}$

The transition diagram of ν' is



(c) From the transition diagram it is evident that the states $\{2\}$, $\{3\}$, $\{2,3\}$ and $\emptyset$ cannot be reached from $\{1\}$. We have $\nu(\{1,2\},\mathtt{a}) = \{1,3\}$, which is an accepting state, and$\nu(\{1\},\mathtt{a}) = \{1\}$, a non-accepting state; thus $\{1\}$ and $\{1,2\}$ cannot be identified. Similarly we have $\nu(\{1,3\},\mathtt{a}) = \{1,2\}$, a non-accepting state but $\nu(\{1,2,3\},\mathtt{a}) = \{1,3\}$, an accepting state; thus $\{1,3\}$ and $\{1,2,3\}$ cannot be identified. Thus a minimal automata is obtained by removing $\{2\}$, $\{3\}$, $\{2,3\}$ from M_B.

Question 8. (*25 points*) Consider the Ackermann function

$$\begin{array}{rcll} \alpha(m,0) & = & m+1 & \text{(i)} \\ \alpha(0,n+1) & = & \alpha(1,n) & \text{(ii)} \\ \alpha(m+1,n+1) & = & \alpha(\alpha(m,n+1),n) & \text{(iii)} \end{array}$$

It has been proved in class that the Ackermann function is monotone in both arguments, i.e., for all i, j, m, and n,

$$\text{if } i < j, \text{ then } \alpha(i,m) < \alpha(j,m) \qquad \text{and} \qquad \text{if } m < n, \text{ then } \alpha(i,m) < \alpha(i,n)$$

and also that it is strictly increasing, i.e., for all m, n

$$\alpha(m,n) > m \qquad \text{and} \qquad \alpha(m,n) > n.$$

(a) Prove that $\alpha(m, n+1) \geq \alpha(m+1, n)$, for all m, n.

(*Hint:* Use monotonicity. The proof is by induction on n, with a secondary induction on m.)

(15 points)

(b) Prove by induction on n that

$$n + n < \alpha(n, 2).$$

(*Hint:* Use part (a).)

(10 points)

Answer. (a) *Base case*: we have

$$\alpha(m,1) \;>\; \alpha(m,0) \;=\; m+1,$$

by monotonicity, hence

$$\alpha(m,1) \;\geq\; m+2 \;=\; \alpha(m+1,0).$$

Inductive step: Suppose $\forall m.\alpha(m, n+1) \geq \alpha(m+1, n)$. To prove $\forall m.\alpha(m, n+2) \geq \alpha(m+1, n+1)$ we use a subsidiary induction on m.

> Subsidiary base case: we have $\alpha(0, n+2) = \alpha(1, n+1)$ by part (ii) of the definition.
>
> Subsidiary inductive step: supposing $\alpha(m, n+2) \geq \alpha(m+1, n+1)$, we have:

$$\begin{array}{rcll} \alpha(m+1,n+2) & = & \alpha(\alpha(m,n+2),n+1) & \text{by def. of } \alpha \\ & \geq & \alpha(\alpha(m+1,n+1),n+1) & \text{by ind.hyp, monotonicity} \\ & > & \alpha(\alpha(m+1,n+1),n) & \text{by monotonicity,} \\ & = & \alpha(m+2,n+1) & \text{by def. of } \alpha \end{array}$$

hence the subsidiary induction is concluded and $\forall m.\alpha(m, n+2) \geq \alpha(m+1, n+1)$ is proved.

Therefore the main induction is also concluded.

(b) We have the *base case*

$$\begin{array}{rcl} \alpha(0,2) & = & \alpha(1,1) \\ & = & \alpha(\alpha(0,1),0) \\ & = & \alpha(0,1)+1 \\ & = & \alpha(1,0)+1 \\ & = & 2 \\ & > & 0 = 0+0 \end{array}$$

and the *inductive step*: assuming $n+n < \alpha(n,2)$ (*inductive hypothesis*)

$$\begin{array}{rcll} \alpha(n+1,2) & = & \alpha(\alpha(n,2),1) & \\ & > & \alpha(n+n,1) & \text{by ind. hyp., monotonicity} \\ & \geq & \alpha(n+n+1,0) & \text{by the } \textit{Fact} \\ & = & n+n+2 = (n+1)+(n+1). & \end{array}$$