University of Cape Town

Department of Computer Science

CSC3005H EXAM

Compilers part 2 & Web based Computing

2006

Marks : 50

Time : 90 minutes

Instructions:

Read the following instructions CAREFULLY before attempting any questions:

- Answer questions from distributed Computing in one book, and questions from Compilers in a *different* book.
- Approximate marks per question are shown in brackets. Please show all your work in arriving at an answer since the reasoning is more important than merely a correct answer.
- Please write the numbers of the questions you have answered on the front cover.
- The use of calculators is permitted.

.

SECTION A: COMPILERS

- Answer **any FIVE** of the seven questions in Section A

Question 1: Symbol Tables [5]

a) A symbol table associates names with attributes. Give one example of an entity whose name could be found in a symbol table. Provide two types of attributes that could be associated with this entity. [3]

Answer:

name of a variable [1]

type of variable [1], size of variable [1], ...

b) Give two examples of static semantics that can be checked with the aid of a symbol table. [2]

Answer:

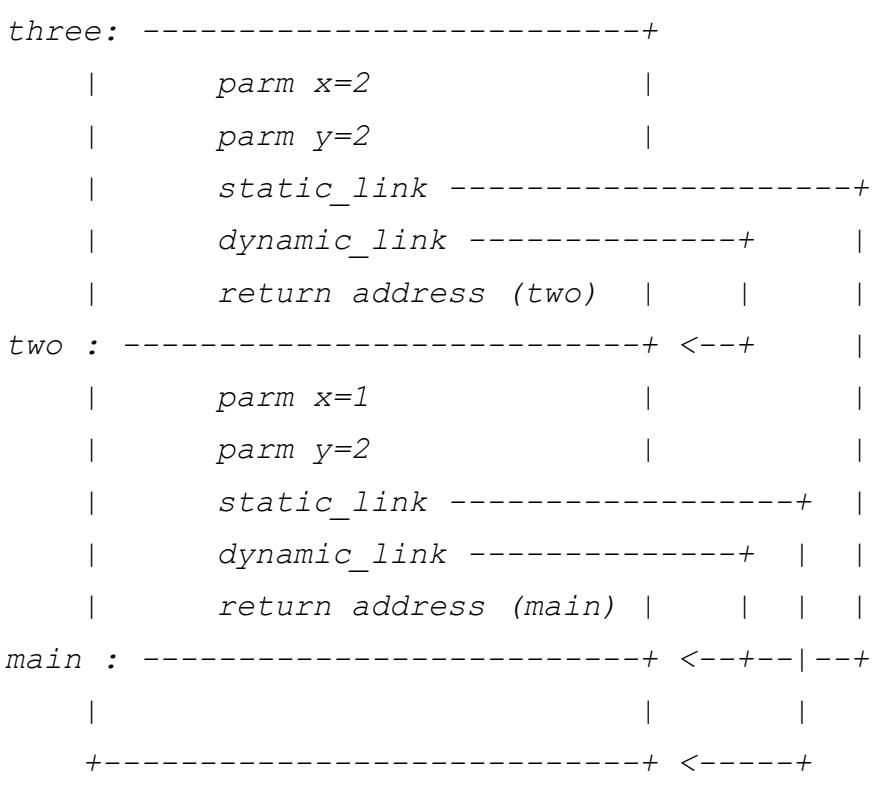
type correspondence [1], array index in bounds [1], declaration before use [1], ...

Question 2: Activation Records [5]

a) Assuming stack-based activation records, draw the full activation record stack corresponding to the function **not_main** at the position marked "%%%", as called by the function **main** in the following program: [5]

```
procedure main
[
   call two(1, 2)
   procedure two (x, y)
   [
      call three(x+1, y)
      procedure three (x, y)
      [
         %%%
         call output (x, y)
      ]
   ]
]
```

Answer:



[Minus one mark for each major error]

Question 3: Intermediate Code [5]

a) Discuss 1 advantage and 1 disadvantage of using intermediate representations. [2]

Answer:

separation of front/back ends [1], easier to apply optimisations to [1]

slower compilation [1], source language not optimally mapped to machine [1]

b) Discuss 3 optimisations that may be applied to IR trees. [3]

Answer:

inlining – replace subprogram calls with body of subprogram [1]

unreachable code elimination – remove code that nevers gets executed [1]

constant propagation – convert uses of a constant name to its value [1]

etc.

Question 4: Basic Blocks [5]

a) What is a basic block? [1]

Answer:

a linear sequence of statements starting with a label and ending with a jump [1]

b) What is a trace? [1]

Answer:

a sequence of basic blocks where each block can jump immediately to the following one [1]

c) For the following program, first separate the code into basic blocks, then rearrange the blocks into traces and finally optimise the code by removing redundant jumps. Show each step separately. [3]

```
Start:  Statement1
        Statement2
        Jump A
B:      Statement3
A:      Statement4
        Jump B
```

Answer:

(a) [1]

```
Start:  Statement1
        Statement2
        Jump A
```

```
B:      Statement3
        Jump A
```

```
A:      Statement4
        Jump B
```

(b) [1]

-

```
Start:  Statement1
        Statement2
        Jump A
A:      Statement4
        Jump B
B:      Statement3
        Jump A
```

(c) [1]

```
Start:  Statement1
```

Statement2

A: *Statement4*

B: *Statement3*

Jump A

Question 5: Instruction Selection [5]

a) Describe the steps of the maximal munch algorithm for instruction selection. [3]

Answer:

find the largest possible tile that matches the root [1]

repeat recursively for the subtrees at the edges of the tile [1]

generate instructions in reverse order [1]

b) In the context of instruction selection by tiling, prove that an optimum tiling (with lowest possible cost) is also optimal (where no two adjacent tiles can be replaced by one with a lower cost than the sum of the two). Hint: use proof by contradiction. [2]

Answer:

Assume that an optimum tiling is not optimal.

Then there exist 2 adjacent tiles than can be replaced with one with a lower cost, so the total cost is not the lowest possible value.

This is a contradiction. Therefore, an optimum tiling MUST be optimal. [2]

Question 6: Liveness Analysis [5]

a) Use the iterative liveness analysis algorithm to calculate the live-in and live-out sets for each of the following statements in a program. Show succ, use, def, out and in sets. Assume the algorithm converges on the first iteration. [5]

```
1: if ( x > 1 )
2:     then y = x * x;
3:     else y = ( 1 / x ) * ( 1 / x );
4: return y+1;
```

Hint: The relevant formulae are:

$$out[n] = \bigcup_{s \in succ[n]} in[s]$$

$$in[n] = use[n] \cup (out[n] - def[n])$$

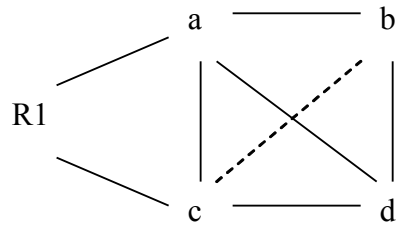
Answer:

Succ [1]	*#*	*Code*	*Use [1]*	*Def [1]*	*Out [1]*	*In [1]*
23	*1*	*If x > 1*	*X*		*X*	*X*
4	*2*	*Y = x^2*	*X*	*Y*	*Y*	*X*
4	*3*	*Y = (1/x)^2*	*X*	*Y*	*Y*	*X*
	4	*Return y+1*	*Y*			*Y*

Question 7: Register Allocation [5]

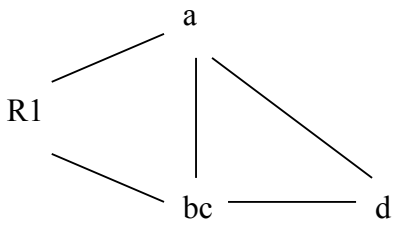
a) Consider the following graph with nodes indicating temporaries and arcs indicating interference. Apply a register colouring algorithm to 3-colour the graph. Assume that R1 is a precoloured node and use George's criterion for conservative coalescing. Clearly show all steps in the algorithm and the final register allocation (R1, R2, R3) to temporaries.

George's criterion: coalesce AB only if all significant-degree neighbours of A interfere with B. [5]



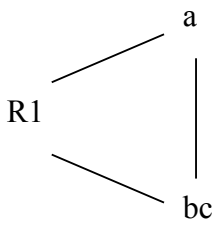
Answer:

No nodes can be simplified, but b and c can be coalesced since each significant degree neighbour of b interferes with c.



[1]

Then, d can be simplified and pushed onto the stack.



[1]

This makes a and bc of <K degree, so they can be simplified as well. [2]

Popping the nodes off the stack, we can then assign

A: R2

B/C: R3

D: R1[1]

SECTION B: WEB BASED COMPUTING

- **Question 8** is compulsory.
- Select and answer **ONE question** from question 9, question 10 and question 11.

Question 8 (Marks 15) ** Marks do not add up (Should it be 5 marks for a and b)

a) Provide a skeleton for the SOAP envelope and its content. [2]

b) Show how you would build the body of a SOAP Message to represent the response to a request for the number of books written by the author "***Iva Rant***" from the database listed below.

Author	Title	Publisher
Iva Rant	One Too Many	My Book
Isabel Necessary	On a bicycle	Tired Books
Iva Rant	All About Me	My Bad Publications
I. Overdunem	How to bake Cakes	Hot Cooking

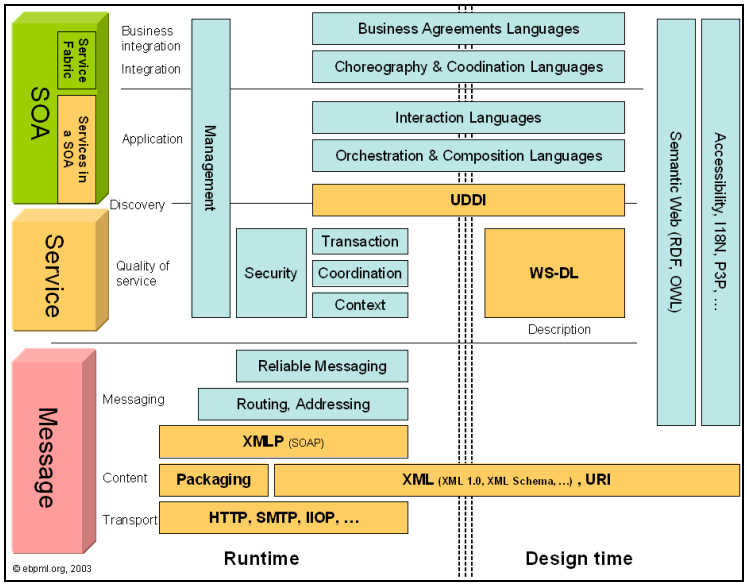
[5]

Answer

```
<?xml version="1.0"?>
<soap:Envelope
 xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
 <soap:Header>
 </soap:Header>
 <soap:Body>
    <soap:Fault>
    </soap:Fault>
 </soap:Body>
</soap:Envelope>
```

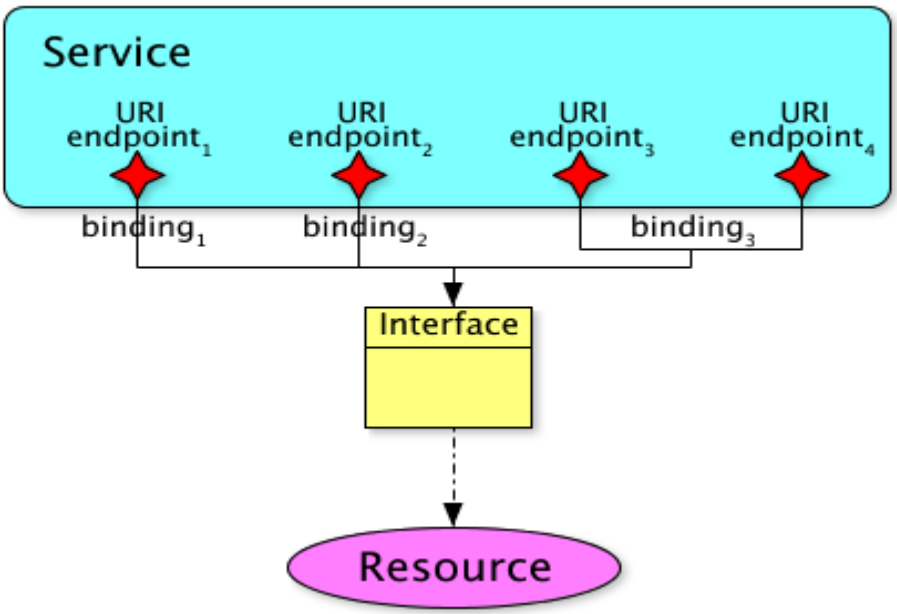
c) Describe the ***Web Services Stack*** show the relationship between the three main levels and the functions performed at each level. [3]

Answer



d) Give a motivation for WSDL and explain its role in the WEB services. [7]

- To see the value of WSDL, imagine you want to start calling a SOAP method provided by one of your business partners.
- You could ask him for some sample SOAP messages and write your application to produce and consume messages that look like the samples.
- However this can be error-prone
- For example, you might see a customer ID of 2837 and assume it's an integer when in fact it's a string.
- WSDL specifies what a request message must contain and what the response message will look like in unambiguous notation.
- WSDL describes Web services starting with the *messages* that are exchanged between the service provider and requester.
- The messages themselves are described abstractly
- They are bound to a concrete network protocol and message format.
- A message consists of a collection of typed data items.



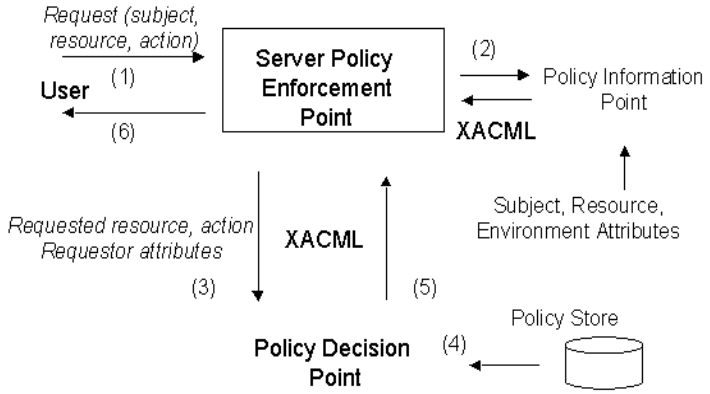
Question 9 (Marks 10)

a) Motivate the need for XACML. Where does it fit into the security structure? [2]

b) Explain how XACML achieves access control through policies and decision. [8]

Answer

- Every enterprise has a need to secure resources accessed by employees, partners, and customers. E.G. browser based access to portals which aggregate resources.
- Clients send requests to servers for resources, but before a server can return that resource it must determine if the requester is authorized to use the resource.
- Allows administrators to define the access control requirements for their application resources.
- The language and schema support include
  - data types,
  - functions,
  - combining logic
- which allow rules to be defined which are used to represent the runtime request for a resource.
-



- When a policy is located which protects a resource, functions compare attributes in the request against attributes contained in the policy rules ultimately yielding a permit or deny decision.
- When a client makes a resource request upon a server, the entity charged with access control by enforcing authorization is called the Policy Enforcement Point.
- In order to enforce policy, this entity will formalize attributes describing the requester at the Policy Information Point and delegate the authorization decision to the Policy Decision Point.
- Applicable policies are located in a policy store and evaluated at the Policy Decision Point, which then returns the authorization decision.
- Using this information, the Policy Enforcement Point can deliver the appropriate response to the client
- The key top-level element is the PolicySet can aggregate other Policy Set elements or Policy elements.
- The Policy element is composed principally of
- Target,
- Rule and
- Obligation elements that are evaluated at the Policy Decision Point to yield and access decision.
- multiple policies may be found applicable to an access decision, a single policy can contain multiple rules. Combining Algorithms are used to reconcile multiple outcomes into a single decision.

Question 10 (Marks 10)

Testing plays an important role in any application and web services is no different. Outline the principles of testing in the Web services environment; describe the areas of concern and strategies to cope with these concerns. [10]

Answer

I would expect a discussion to cover most of the points below.

- Intranet and Internet web services provides subtly different problems.
- With an Intranet web service you, as an organization, are likely to have control over who accesses your web to a theoretical maximum.
- Similarly, you can make certain assumptions about security.
- With an Internet web service, anybody can access it.
- This means that there are additional scalability and security considerations.
- they do not display a user interface that can be tested.
- This means that they hard to test manually, but are an ideal candidate for automated testing
- programming skills are almost certainly needed for testers who need to test web services.
- A web service is not the sort of application you can test by key-bashing.

Does the service function as expected?

- If you have a web service that divides two numbers, does it give the expected result?
- If you pass in a 0 as the denominator, does it handle this correctly?
- Does your web service implement security/authentication as it should?
- Does your web service support all the communications protocols it is meant to?

- Because your web service can be accessed by clients that you can't control, what happens if they make requests you aren't expecting?
- Bounds testing and error checking are especially important.
- Load/stress testing how does your web service scales as the number of clients accessing it increases?
- What you need to know now is if it will cope with 10, 100 or 1,000 users, or how many users it will cope with.
- If you double the number of users, do response times stay the same?
- If you double the number of servers running your web service, does its capacity double?
- Once your web service is live and being used by real clients, you will need to keep an eye on it.
- Is it still working?
- Are response times adequate?
- At what times of day is it busiest?
- It is essential to monitor the web service.

Question 11 (Marks 10)

Part of the Web Services stack provides functionality for Orchestration and Choreography. What is the descriptive name given to this part of the stack? Give an overview of the services included in the section; in particular what is the relationship between WSFL, WSCI, and BPEL. Give examples of the way in which they function. [10]

Answer

I expect a discussion to cover some of the points below.

Different models offered:

- Microsoft's XLang,
- IBM's Web Services Flow Language (WSFL), and the
- Business Process Management Initiative's Business Process Modeling Language (BPML)
- focus on the nuts and bolts of passing business-oriented messages back and forth.
- WSFL - This is the model of interaction prevalent even in simple business transactions, both in the business-to-consumer and (especially) in the business-to-business scenarios.
- Even a process as simple as ordering merchandise online involves at least three parties—a customer, an e-commerce retailer, and a shipper—as well as direct interactions between the three of them.
- In WSFL, flow models are used for the specification of complex interactions of services.
- A flow model describes a usage pattern of a collection of available services,
- so that the composition of those services provides the functionality needed to achieve a certain business goal
- The flow model specifies precisely how those services (steps in the flow) are combined,
- specifying in which order they have to be performed (including the possibility of concurrent execution),
- the conditional logic deciding whether an individual step or a group of steps have to be performed at all or should be looped, and the passing of data between the involved steps.
-

- The WS-CDL XML-based language "describes peer-to-peer collaborations of Web Services participants by defining, from a global viewpoint, their common and complementary observable behavior, where ordered message exchanges result in accomplishing a common business goal.
- According to the W3C announcement, the Web Services Choreography Description Language is a "necessary complement to end point languages such as BPEL and Java. WS-CDL provides them with the global model they need to ensure that end point behavior ? the 'rules of engagement' ? Is consistent across cooperating services.
- WS-CDL describes the set of rules that explains how different components may act together, and in what sequence, giving a flexible systemic view of the process.
- It serves as a necessary complement to BPEL, Java, and other programming languages which describe one endpoint on a transaction, rather than the full system.
- Choreography Speeds Time to Market and
- Reduces Cost of Ownership
- One of the aims of Web services is integration (combining components into a system) to reduce connectivity costs and increase the utility and thus the value of information.
- New Standards:
- BPEL4W, Business Process Execution Language for Web services
- The typical scenario is that there is a message received into the BPEL process.
- The process may then invoke a series of external services to gather additional data, and then respond to the requestor in some fashion.
- In the figure below, the <receive>, <reply>, and <invoke>
- messages all represent basic activities for connecting the services together.