# Questions about Parallelizing Compilers

1. Why do parallelizing compilers focus on loops?
2. How many iterations does it take for a loop with a single statement to beneficially run in parallel?
3. On what programs are autoparallelizers most successful? Why?
4. On what programs are autoparallelizers least successful? Why?
5. What was the success rate of the first generation of autoparallelizers for science/engineering applications?
6. What techniques were most important for automatic parallelization in the first generation of such compilers?
7. Why is a flow dependence difficult to eliminate? What does it take to eliminate it?
8. What dependences does the privatization transformation eliminate? How does it do it?
9. What dependences do you see in the loop

for (j=1; j<n; j++) {

sum = sum + a[j]

}

1. How does loop interchange improve performance?
2. What is an affine subscript?
3. Why does the range test need to determine monotonicity? How does it do it?
4. What is a dependence distance?
5. Write the data dependence problem (system of equations and inequalities) for the following loop, to test for dependence between the two references to array A.

DO i=1,n

DO j=1,m

A(i\*3+2,j-5+i\*2+7) = …

… = A(i,j)

ENDDO

ENDDO

1. Why are there so many data dependence tests?
2. Give an intuition why privatization is the most important parallelization technique.
3. What make array privatization more difficult than scalar privatization?
4. Why was scalar expansion invented before scalar privatization?
5. Does privatization come with overheads? Which ones?
6. What is the relationship between induction variable substitution and strength reduction?
7. What makes the substitution of generalized induction variables difficult?
8. Describe the overheads caused by reduction parallelization.
9. A reduction statement looks similar to an induction statement. What is the difference?
10. What is the purpose of loop skewing?
11. When would you do loop fusion and when loop distribution?
12. Write a correct and an incorrect direction vector
13. Which two transformations create loop blocking? In what sequence?
14. What technique creates spatial locality and what technique achieves temporal locality?
15. In terms of parallelization techniques, how do vector machines differ from multiprocessors?
16. Why is interprocedural analysis important?
17. How can inlining help interprocedural analysis? What are some of the drawbacks?
18. What issues exist when optimizing for distributed-memory architectures ?
19. What makes automatic data distribution intrinsically difficult?
20. What is the owner-computes scheme?
21. What was the goal of HPF?
22. Explain the basic translation scheme of the OpenMP-to-MPI compilation method
23. Give an example of two consecutive parallel regions that access an array A. Compute the array sections and communication sets.