Bernstein's Conditions for Parallel Execution

Introduction

In parallel computing, it is essential to determine whether two instructions can be executed simultaneously without causing data conflicts. **Bernstein's Conditions** provide a set of criteria to ensure that two instructions do not interfere with each other when run in parallel. These conditions define data dependencies between instructions and help identify if they can be executed concurrently.

Definitions

Let I be an instruction in a program. We denote:

- $R(I)$: the set of variables read by the instruction I (variables that remain unchanged during the execution of I).
- $W(I)$: the set of variables written to by the instruction I (variables that are modified by the execution of I).

Bernstein's Conditions

Two instructions, I_1 and I_2, can be executed in parallel if the following conditions are satisfied:

1. **Condition 1:** $R(I_1) \cap W(I_2) = \emptyset$

   This condition ensures that instruction I_1 does not read any variable that is modified by instruction I_2. If this condition is violated, I_1 would be reading potentially incorrect data if I_2 is executed simultaneously.

2. **Condition 2:** $W(I_1) \cap R(I_2) = \emptyset$

   This condition ensures that instruction I_1 does not modify any variable that is read by instruction I_2. If this condition is violated, I_2 could read an inconsistent value if I_1 is executed in parallel.

3. **Condition 3:** $W(I_1) \cap W(I_2) = \emptyset$

   This condition ensures that I_1 and I_2 do not modify the same variable. If they do, there could be conflicting changes to that variable if both instructions execute at the same time.

If all three conditions are satisfied, then I_1 and I_2 can safely execute in parallel without any data conflicts.

Example

Consider two instructions I_1 and I_2 with the following read and write sets:

- $R(I_1) = \{a, b\}$
- $W(I_1) = \{c\}$
- $R(I_2) = \{c\}$
- $W(I_2) = \{d\}$

Let us check if I_1 and I_2 can execute in parallel by verifying Bernstein's Conditions:

- Condition 1: $R(I_1) \cap W(I_2) = \{a, b\} \cap \{d\} = \emptyset$, which is satisfied.
- Condition 2: $W(I_1) \cap R(I_2) = \{c\} \cap \{c\} = \emptyset$, which is **not satisfied**.
- Condition 3: $W(I_1) \cap W(I_2) = \{c\} \cap \{d\} = \emptyset$, which is satisfied.

Since Condition 2 is not satisfied, I_1 and I_2 cannot execute in parallel without risking data conflicts.

Conclusion

Bernstein's Conditions provide a systematic approach to determine data independence and ensure safe parallel execution of instructions. By verifying these conditions, we can identify which instructions can run concurrently, optimizing the execution of a program and reducing potential data conflicts.