Interference Freedom

- In the last lecture, we learned that threads that *are pairwise disjoint* and *have disjoint conditions* don't interfere with each other, and we can apply the parallelism rule to prove the correctness of their parallel composition. We also learned that *being pairwise disjoint* and *having disjoint conditions* are too strict for *being interference free*.
- So that we can use the parallelism rule to prove more parallel statements, we want to find out what makes a thread not interfere the execution of another thread.

1. Let's look at the two following pairs of threads.
   a) We can apply the parallelism rule on these two threads:

$$\underline{\{x > 0\}}\ \underline{x \coloneqq x + 1}\ \underline{\{x > 1\}}$$
$$\underline{\{x > 0\}}\ \underline{x \coloneqq x + 2}\ \underline{\{x > 2\}}$$

   b) We cannot do the same for these two threads.

$$\underline{\{x = 0\}}\ x \coloneqq x + 1\ \underline{\{x = 1\}}$$
$$\{x = 0\}\ \underline{x \coloneqq x + 2}\ \{x = 2\}$$

   What is the difference?
   In a), given the precondition of the second thread $x > 0$ being true, the execution of $x \coloneqq x + 1$ in the first thread doesn't change the correctness of $x > 0$; similarly, given the postcondition of the second thread $x > 2$ being true, the execution of $x \coloneqq x + 1$ in the first thread doesn't change the correctness of $x > 2$.

   Also in a), given the precondition of the first thread $x > 0$ being true, the execution of $x \coloneqq x + 2$ in the second thread doesn't change the correctness of $x > 0$; similarly, given the postcondition of the first thread $x > 1$ being true, the execution of $x \coloneqq x + 2$ in the second thread doesn't change the correctness of $x > 1$.

   The same statements don't hold for the threads in b).

- From the above observations, we define the **interference-freedom check** for statement $\{p\}\ S\ \{\dots\}$ versus a predicate q is the triple $\{p \wedge q\}\ S\ \{q\}$: if this triple is totally valid, then we say that $\{p\}\ S\ \{\dots\}$ **does not interfere with** q.
  - Similar to the "disjointedness test" and "disjoint condition test", this interference-freedom check is also static and can be too strict.

2. Consider the following pairs of statement $\{p\}\ S\ \{\dots\}$ and predicate q, does the statement interfere with q?

   a. $\{p\}\ x \coloneqq x + 1\ \{\dots\}$ and $x \geq 0$.
   No. The triple we need to check its validity is $\{p \wedge x \geq 0\}\ x \coloneqq x + 1\ \{x \geq 0\}$, and it is totally valid.

   b. $\{p\}\ x \coloneqq x + 1\ \{\dots\}$ and $x < 0$.
   Yes. The triple we need to check its validity is $\{p \wedge x < 0\}\ x \coloneqq x + 1\ \{x < 0\}$, it is not valid since state $\{x = -1\}$ doesn't satisfy it.

   c. $\{p\}\ S\ \{\dots\}$ and F.
   No. The triple we need to check its validity is $\{p \wedge F\}\ S\ \{F\}$, and it is totally valid.

d. $\{x \% 4 = 2\}\ x \coloneqq x + 4\ \{\dots\}$ and $x \% 2 = 0$.
No. The triple we need to check its validity is $\{x \% 4 = 2 \land x \% 2 = 0\}\ x \coloneqq x + 4\ \{x \% 2 = 0\}$, and it is totally valid.

- For threads $\{p_1\}\ S_1\ \{q_1\}$ and $\{p_2\}\ S_2\ \{q_2\}$, what are the interference-freedom check we need to decide their interference freedom?
  - Someone may say the four following checks are enough:
    $\{p_1\}\ S_1\ \{\dots\}$ vs p_2, thus the triple $\{p_1 \land p_2\}\ S_1\ \{p_2\}$
    $\{p_1\}\ S_1\ \{\dots\}$ vs q_2, thus the triple $\{p_1 \land q_2\}\ S_1\ \{q_2\}$
    $\{p_2\}\ S_2\ \{\dots\}$ vs p_1, thus the triple $\{p_2 \land p_2\}\ S_1\ \{p_1\}$
    $\{p_2\}\ S_2\ \{\dots\}$ vs q_1, thus the triple $\{p_2 \land q_1\}\ S_1\ \{q_1\}$

    They are probably not enough. The problem is that: what if S_1 is a long statement and we execute S_1 for several steps then switch to S_2 then back to S_1? Do we need to consider other pre- and post- conditions in the proof outline of S_1? Let's look at the following example.

3. Let $S \equiv [x \coloneqq y + y \parallel x \coloneqq y; x \coloneqq x + y]$ and we execute S in a state $\{y = \alpha\}$ (which satisfies the precondition of the program). The postcondition we need for this program $x = 2\alpha$. Are there any race conditions in this program?

   - We can get the following evaluation graph:

$$\begin{tikzcd}
 & & \langle [x \coloneqq y + y \parallel x \coloneqq y; x \coloneqq x + y], \{y = \alpha\}\rangle \arrow[dll] \arrow[dr] & & \\
\langle [E \parallel x \coloneqq y; x \coloneqq x + y], \{y = \alpha, x = 2\alpha\}\rangle \arrow[d] & & & \langle [x \coloneqq y + y \parallel x \coloneqq x + y], \{y = \alpha, x = \alpha\}\rangle \arrow[dl] \arrow[dr] & \\
\langle [E \parallel x \coloneqq x + y], \{y = \alpha, x = \alpha\}\rangle \arrow[dr] & & \langle [x \coloneqq y + y \parallel E], \{y = \alpha, x = 2\alpha\}\rangle \arrow[dl] & & \langle [E \parallel x \coloneqq x + y], \{y = \alpha, x = 2\alpha\}\rangle \arrow[d] \\
 & \langle [E \parallel E], \{y = \alpha, x = 2\alpha\}\rangle & & & \langle [E \parallel E], \{y = \alpha, x = 3\alpha\}\rangle
\end{tikzcd}$$

   - We can see that $M(S, \{y = \alpha\}) = \big\{\{y = \alpha, x = 2\alpha\}, \{y = \alpha, x = 3\alpha\}\big\}$, and $\{y = \alpha, x = 3\alpha\}$ doesn't satisfy the postcondition we need. However, if we look at these two threads separately, we see that $x \coloneqq y + y$ and $x \coloneqq y; x \coloneqq x + y$ can both give us $\{y = \alpha, x = 2\alpha\}$.
   - The invalidity here is caused by **interleaving** a thread: we execute one thread for some steps then switch to another thread before finishing the execution. The execution orders of threads causes this invalidity, so the program has race conditions.

- For different types of statement S as a thread, when can "interleaving thread S" occur?
  - An assignment statement or a $\mathbf{skip}$ statement is atomic itself; we cannot interleave it.
  - Interleaving a sequential statement can occur after the execution of one statement and before another one in the sequential.
  - For $\mathbf{if} - \mathbf{else}$ and $\mathbf{while}$ statements, interleaving the execution of loop body or the execution of a branch can occur and can be harmful. Also, although evaluation of a Boolean expression is atomic, interleaving can occur between inspection of the result and the jump to the next configuration.

        - For example, if $\sigma(x) = 0$, then $\langle \mathbf{if}\ x = 0\ \mathbf{then}\ S_1\ \mathbf{else}\ S_2\ \mathbf{fi},\ \sigma\rangle \to \langle S_1,\ \sigma\rangle$. In parallel, another statement can be executed between the if test and the start of the true branch. It is possible that we have $\langle[\mathbf{if}\ x = 0\ \ \mathbf{then}\ S_1\ \mathbf{else}\ S_2\ \mathbf{fi} \parallel x \coloneqq x + 1\ ],\ \sigma\rangle \to \langle[S_1 \parallel x \coloneqq x + 1],\ \sigma\ \rangle \to \langle[\ S_1 \parallel\ \ E\ ],$ $\sigma[x \mapsto 1]\rangle$. In this last configuration, we're about to execute S_1 not in σ, but in $\sigma[x \mapsto 1]$. This is a problem because if x is now evaluated to 1 and the condition $x = 0$ is not true anymore and we supposed to execute S_2 instead.

- Are all the interleaving harmful? Let's look at this example.

4. Does the program $\{x \geq 10\}\ [x \coloneqq x + 1;\ x \coloneqq x + 2 \parallel x \coloneqq x + 3]\ \{x \geq 16\}$ have race conditions?
    - No, no matter whether we interleave the first thread or not, we always increase x's value by 6 after the execution.

- The above example shows that not all the interleaving is harmful. Informally, we define a part of the thread that must be executed atomically (no interleaving) to get the expected postcondition as a **critical section**. The race conditions caused by interleaving a critical section is called a **critical section problem**.
    - For example, in the program in Example 3, "$x \coloneqq y; x \coloneqq x + y$" should be a critical section.
    - In the program in Example 4, there are no critical sections.

- To decide whether an interleaving is harmful needs us to analyze the semantic of the parallel program.
- To avoid critical section problems, we can clarify critical sections in threads by defining the **atomic regions**:
    - If S is a statement, then $< S >$ is an **atomic region statement (or atomic statement in short)** with body S. The evaluation of S is considered as one step: If $\langle S, \sigma\rangle \to^k \langle E, \tau\rangle$ and $k \geq 1$ then $\langle < S >, \sigma\rangle \to \langle E, \tau\rangle$.

- Using atomic regions gives us control of the occurrence of possible interleaving; however, making more or larger atomic sections is no panacea: The more or larger atomic regions code has, the less interleaving and hence parallelism we have.

Now, let's come back to interference-freedom checks for threads $\{p_1\}\ S_1\ \{q_1\}$ and $\{p_2\}\ S_2\ \{q_2\}$. Since we are considering possibilities of interleaving threads, we need to consider the proof outlines of these threads. Let's denote the (partial or full) proof outline of $\{p_i\}\ S_i\ \{q_i\}$ as $\{p_i\}\ S_i^*\ \{q_i\}$.

- **The (atomic) statement $\{p_1\}\ T_1\ \{\dots\}$ does not interfere with the proof outline** $\{p_2\}\ S_2^*\ \{q_2\}$, if $\{p_1\}\ T_1\ \{\dots\}$ doesn't interfere with $\underline{p_2}$ nor with $\underline{q_2}$, nor with any precondition before an atomic statement in S_2^* (i.e. any r where $\underline{\{r\}} < \cdots > \{\dots\}$ appears in S_2^*).
- **A proof outline** $\{p_1\}\ S_1^*\ \{q_1\}$ **does not interfere with the proof outline** $\{p_2\}\ S_2^*\ \{q_2\}$, if every atomic statement $\{r\}\ T_1\ \{\dots\}$ in S_1^* doesn't interfere with $\{p_2\}\ S_2^*\ \{q_2\}$.
- Two proof outlines $\{p_1\}\ S_1^*\ \{q_1\}$ and $\{p_2\}\ S_2^*\ \{q_2\}$ are **interference free** if they don't interfere with each other.

5. How do we decide whether statement $\underline{\{x > 0\}\ x \coloneqq x + 1\ \{x > 1\}}$ interfere with the following thread (written in a proof outline)?

```
$S^* \equiv$
$\underline{\{x > 0\}}\ y \coloneqq 5; \underline{\{x > 0 \wedge y = 5\}}\ z \coloneqq 0;\ \{x > 0 \wedge y = 5 \wedge z = 0\}$
$\{\mathbf{inv}\ \underline{x + z \geq 5}\}$
$<\ \mathbf{while}\ (x > y)\ \mathbf{do}$
    $\{x + z \geq 5 \wedge x > y\}$
    $x \coloneqq x - 1; z \coloneqq z + 1\ \{x + z \geq 5\}$
$\mathbf{od}\ >$
$\underline{\{x + z \geq 5 \wedge x \leq y\}}$
```

    - We need to do the interference-freedom check of statement $\{x > 0\}\ x \coloneqq x + 1\ \{\ldots\}$ vs the following predicates:
        - $x > 0$, thus check the validity of triple $\{x > 0\}\ x \coloneqq x + 1\ \{x > 0\}$. ✓
        - $x > 0 \land y = 5$, thus check the validity of triple $\{x > 0 \land y = 5\}\ x \coloneqq x + 1\ \{x > 0 \land y = 5\}$. ✓
        - $x + z \geq 5$, thus check the validity of triple $\{x > 0 \land x + z \geq 5\}\ x \coloneqq x + 1\ \{x + z \geq 5\}$. ✓
        - $x + z \geq 5 \land x \leq y$, thus check the validity of triple $\{x > 0 \land x + z \geq 5 \land \underline{x \leq y}\}\ x \coloneqq x + 1\ \{x + z \geq 5 \land \underline{x \leq y}\}$. X

      These predicates are either the very first precondition, the very last postcondition of S^*, or they are the precondition of some atomic statement in S^*.

- Why don't we check whether a statement interferes with the postcondition of an atomic region or the pre- and post- condition of a non-atomic statement?
    - Interleaving is allowed after the execution of an atomic region and in non-atomic regions. As long as a statement does not interfere with the precondition of an atomic statement, we can switch back to this thread and execute the atomic region without problems.

6. Are the following threads interference free? Here predicate function $even(x) \equiv x \,\%\, 2 = 0$

   $S_1^* \equiv \{even(x)\}\ x \coloneqq x;\ \{even(x)\}\ x \coloneqq x + 2\ \{even(x)\}$

   $S_2^* \equiv \{x > 2\}$ ✓ **if** $even(x)$ **then**

   $\quad \{x > 2 \land even(x)\}$ ✓ $x \coloneqq x - 2\ \{x > 0 \land even(x)\}$

   $\quad$ **else** $\{x > 2 \land \neg even(x)\}$ ✓ $x \coloneqq x - 1\ \{x > 1 \land even(x)\}$ **fi**

   $\quad \{x > 0\}$ ✓

    - To decide whether S_2^* interferes with S_1^*, we need to check the validity of the following triples:
        - $\{x > 2 \land even(x)\}\ x \coloneqq x - 2\ \{even(x)\}$. It is totally valid.
        - $\{x > 2 \land \neg even(x) \land even(x)\}\ x \coloneqq x - 1\ \{even(x)\}$. It is totally valid.

    - To decide whether S_1^* interferes with S_2^*, we need to check the validity of the following triples:
        - $\{even(x) \land x > 2\}\ x \coloneqq x\ \{x > 2\}$. It is totally valid.
        - $\{even(x) \land x > 2 \land even(x)\}\ x \coloneqq x\ \{x > 2 \land even(x)\}$. It is totally valid.
        - $\{even(x) \land x > 2 \land \neg even(x)\}\ x \coloneqq x\ \{x > 2 \land \neg even(x)\}$. It is totally valid.
        - $\{even(x) \land x > 0\}\ x \coloneqq x\ \{x > 0\}$. It is totally valid.
        - $\{even(x) \land x > 2\}\ x \coloneqq x + 2\ \{x > 2\}$. It is totally valid.
        - $\{even(x) \land x > 2 \land even(x)\}\ x \coloneqq x + 2\ \{x > 2 \land even(x)\}$. It is totally valid.
        - $\{even(x) \land x > 2 \land \neg even(x)\}\ x \coloneqq x + 2\ \{x > 2 \land \neg even(x)\}$. It is totally valid.
        - $\{even(x) \land x > 0\}\ x \coloneqq x + 2\ \{x > 0\}$. It is totally valid.

    - Thus, $\{p_1\}\ S_1^*\ \{q_1\}$ and $\{p_2\}\ S_2^*\ \{q_2\}$ are interference free.

Now, we can update the Parallelism Rule:

- **Parallelism Rule**

  For $k = 1 \ldots n$, if proof outlines $\{p_k\}\ S_k^*\ \{q_k\}$ are interference free:

  1 $\{p_1\}\ S_1^*\ \{q_1\}$

  2 $\{p_2\}\ S_2^*\ \{q_2\}$

  …

  n $\{p_n\}\ S_n^*\ \{q_n\}$

  n + 1 $\{p_1 \land p_2 \land \ldots \land p_n\}\ [S_1^* \parallel S_2^* \parallel \cdots \parallel S_n^*]\ \{q_1 \land q_2 \land \ldots \land q_n\}$ parallelism 1,2, … , n