Semantics and typing of break and continue

Programming Paradigms · Session 10 · lab · **with solutions**

The language WHILE of block 2

Expressions	$e ::= n \mid b \mid x \mid e\ op\ e$
Statements	$s ::= x \leftarrow e \mid \mathsf{if}\ e\ \mathsf{then}\ s\ \mathsf{else}\ s \mid \mathsf{while}\ e\ \mathsf{do}\ s \mid s; s \mid \mathsf{skip}$
Values	$v ::= n \mid b$
Types	$\tau ::= \mathsf{int} \mid \mathsf{bool}$

Here n is an integer constant, $b \in \{\mathsf{true}, \mathsf{false}\}$ and $op \in \{+, -, \times, <, >, =\}$. An environment E is a finite map from variables to values, and a typing environment Γ a finite map from variables to types.

Big-step semantics. $E, e \twoheadrightarrow v$: in the environment E, the expression e has the value v. $E, s \twoheadrightarrow E'$: in the environment E, the run of the statement s ends, in the environment E'.

$$\frac{}{E, n \twoheadrightarrow n}\ (\textsc{Int}) \qquad \frac{}{E, b \twoheadrightarrow b}\ (\textsc{Bool}) \qquad \frac{x \in \mathrm{dom}(E)}{E, x \twoheadrightarrow E(x)}\ (\textsc{Var})$$

$$\frac{E, e_1 \twoheadrightarrow n_1 \quad E, e_2 \twoheadrightarrow n_2 \quad n = n_1 + n_2}{E, e_1 + e_2 \twoheadrightarrow n}\ (\textsc{Plus})$$

$$\frac{}{E, \mathsf{skip} \twoheadrightarrow E}\ (\textsc{Skip}) \qquad \frac{E, e \twoheadrightarrow v}{E, x \leftarrow e \twoheadrightarrow E\{x \mapsto v\}}\ (\textsc{Assign}) \qquad \frac{E, s_1 \twoheadrightarrow E_1 \quad E_1, s_2 \twoheadrightarrow E_2}{E, s_1; s_2 \twoheadrightarrow E_2}\ (\textsc{Seq})$$

$$\frac{E, e \twoheadrightarrow \mathsf{true} \quad E, s_1 \twoheadrightarrow E_1}{E, \mathsf{if}\ e\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2 \twoheadrightarrow E_1}\ (\textsc{IfTrue}) \qquad \frac{E, e \twoheadrightarrow \mathsf{false} \quad E, s_2 \twoheadrightarrow E_2}{E, \mathsf{if}\ e\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2 \twoheadrightarrow E_2}\ (\textsc{IfFalse})$$

$$\frac{E, e \twoheadrightarrow \mathsf{false}}{E, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow E}\ (\textsc{WhileFalse})$$

$$\frac{E, e \twoheadrightarrow \mathsf{true} \quad E, s \twoheadrightarrow E_1 \quad E_1, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow E_2}{E, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow E_2}\ (\textsc{WhileTrue})$$

Typing. $\Gamma \vdash e : \tau$: in the typing environment Γ, the expression e has the type τ. $\Gamma \vdash s$: in the typing environment Γ, the statement s is well typed.

$$\frac{}{\Gamma \vdash n : \mathsf{int}}\ (\text{T-}\textsc{Int}) \qquad \frac{}{\Gamma \vdash b : \mathsf{bool}}\ (\text{T-}\textsc{Bool}) \qquad \frac{x \in \mathrm{dom}(\Gamma)}{\Gamma \vdash x : \Gamma(x)}\ (\text{T-}\textsc{Var})$$

$$\frac{\Gamma \vdash e_1 : \mathsf{int} \quad \Gamma \vdash e_2 : \mathsf{int}}{\Gamma \vdash e_1 + e_2 : \mathsf{int}}\ (\text{T-}\textsc{Plus}) \qquad \frac{\Gamma \vdash e_1 : \mathsf{int} \quad \Gamma \vdash e_2 : \mathsf{int}}{\Gamma \vdash e_1 < e_2 : \mathsf{bool}}\ (\text{T-}\textsc{Less})$$

$$\frac{}{\Gamma \vdash \mathsf{skip}}\ (\text{T-}\textsc{Skip}) \qquad \frac{\Gamma \vdash s_1 \quad \Gamma \vdash s_2}{\Gamma \vdash s_1; s_2}\ (\text{T-}\textsc{Seq}) \qquad \frac{\Gamma \vdash x : \tau \quad \Gamma \vdash e : \tau}{\Gamma \vdash x \leftarrow e}\ (\text{T-}\textsc{Assign})$$

$$\frac{\Gamma \vdash e : \mathsf{bool} \quad \Gamma \vdash s_1 \quad \Gamma \vdash s_2}{\Gamma \vdash \mathsf{if}\ e\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2}\ (\text{T-}\textsc{If}) \qquad \frac{\Gamma \vdash e : \mathsf{bool} \quad \Gamma \vdash s}{\Gamma \vdash \mathsf{while}\ e\ \mathsf{do}\ s}\ (\text{T-}\textsc{While})$$

Conventions. The operators $-$ and $\times$ have the rules of $+$, (PLUS) and (T-PLUS), each with its own operation on integers. The operators $<$, $>$ and $=$ have the typing rule (T-LESS), and an evaluation rule of the form of (PLUS) whose value is the Boolean true when the comparison of n_1 and n_2 holds, and false otherwise. The environment $E\{x \mapsto v\}$ maps x to v and every other variable as E does. The sequence associates to the right: $s_1; s_2; s_3$ is $s_1; (s_2; s_3)$. A branch or a loop body that is itself a sequence stands in parentheses.

A *derivation* of a judgment is a tree: the judgment stands at the root, every node is the conclusion of an instance of one of the rules, and the node's children are the premises of that instance. Each step of a derivation carries the name of its rule.

The extension: two statements and three outcomes

The lab extends WHILE with two statements:

$$s \ ::= \ \cdots \mid \mathsf{break} \mid \mathsf{continue}$$

The statement break ends the loop whose body is running: the run goes on after that loop. The statement continue ends the current run of the body: the run goes on with the next evaluation of the guard of that loop. Expressions, values and environments are unchanged, and so is the judgment $E, e \twoheadrightarrow v$.

Both statements interrupt the statement that contains them. The judgment for statements therefore records how a run ends, with an *outcome* ω:

$$E, s \twoheadrightarrow (E', \omega) \qquad\qquad \omega \ ::= \ \mathsf{normal} \mid \mathsf{broken} \mid \mathsf{continued}$$

It reads: in the environment E, the run of s ends, in the environment E' and with the outcome ω. The outcome broken says that the run of s ended on a break that no loop inside s has caught, and continued says the same of a continue. An outcome is a part of the judgment, not a value of the language.

Four rules of the extended semantics are given. They are the rules (SKIP), (ASSIGN), (IFTRUE) and (IFFALSE) of block 2, with an outcome:

$$\frac{}{E, \mathsf{skip} \twoheadrightarrow (E, \mathsf{normal})}\ (\textsc{Skip}) \qquad \frac{E, e \twoheadrightarrow v}{E, x \leftarrow e \twoheadrightarrow (E\{x \mapsto v\}, \mathsf{normal})}\ (\textsc{Assign})$$

$$\frac{E, e \twoheadrightarrow \mathsf{true} \quad E, s_1 \twoheadrightarrow (E_1, \omega)}{E, \mathsf{if}\ e\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2 \twoheadrightarrow (E_1, \omega)}\ (\textsc{IfTrue}) \qquad \frac{E, e \twoheadrightarrow \mathsf{false} \quad E, s_2 \twoheadrightarrow (E_2, \omega)}{E, \mathsf{if}\ e\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2 \twoheadrightarrow (E_2, \omega)}\ (\textsc{IfFalse})$$

The exercises of the first part build the other rules: those of break, of continue, of the sequence and of the loop.

Big-step rules of the extension

Exercise 1 Rules of break and continue

Given the judgment $E, s \twoheadrightarrow (E', \omega)$, write the rule of break and the rule of continue. Each of the two is an axiom, a rule without a premise.

Solution.

$$\frac{}{E, \mathsf{break} \twoheadrightarrow (E, \mathsf{broken})}\ (\textsc{Break}) \qquad \frac{}{E, \mathsf{continue} \twoheadrightarrow (E, \mathsf{continued})}\ (\textsc{Continue})$$

Neither statement changes the environment; each ends at once, with the outcome that carries its name.

Exercise 2 Rules of the sequence

Given a sequence $s_1; s_2$, write its three rules, one for each outcome of the run of s_1. In each rule the run of s_1 is a premise.

Solution.

$$\frac{E, s_1 \twoheadrightarrow (E_1, \mathsf{normal}) \quad E_1, s_2 \twoheadrightarrow (E_2, \omega)}{E, s_1; s_2 \twoheadrightarrow (E_2, \omega)}\ (\textsc{SeqNormal})$$

$$\frac{E, s_1 \twoheadrightarrow (E_1, \mathsf{broken})}{E, s_1; s_2 \twoheadrightarrow (E_1, \mathsf{broken})}\ (\textsc{SeqBreak}) \qquad \frac{E, s_1 \twoheadrightarrow (E_1, \mathsf{continued})}{E, s_1; s_2 \twoheadrightarrow (E_1, \mathsf{continued})}\ (\textsc{SeqContinue})$$

In (SEQNORMAL) the outcome of the sequence is the outcome ω of s_2, whatever it is: the rule

also covers a sequence whose second statement ends on a break or a continue. In (SeqBreak) and (SeqContinue) the statement s_2 stands in no premise: it does not run, and the outcome of s_1 is the outcome of the sequence.

Exercise 3 Derivation of a run that ends on a break

Given the environment $\{x \mapsto 0\}$ and the statement

$$x \leftarrow 1;\ \mathsf{break};\ x \leftarrow 2,$$

write the derivation of its run with the given rules and the rules of the two exercises above, and write the final environment and the outcome.

Solution. The statement is $x \leftarrow 1; (\mathsf{break}; x \leftarrow 2)$. With $E_0 = \{x \mapsto 0\}$ and $E_1 = \{x \mapsto 1\}$:

$$\dfrac{\dfrac{\dfrac{}{E_0, 1 \twoheadrightarrow 1}\,(\textsc{Int})}{E_0, x \leftarrow 1 \twoheadrightarrow (E_1, \mathsf{normal})}\,(\textsc{Assign}) \qquad \dfrac{\dfrac{}{E_1, \mathsf{break} \twoheadrightarrow (E_1, \mathsf{broken})}\,(\textsc{Break})}{E_1, \mathsf{break}; x \leftarrow 2 \twoheadrightarrow (E_1, \mathsf{broken})}\,(\textsc{SeqBreak})}{E_0, x \leftarrow 1; \mathsf{break}; x \leftarrow 2 \twoheadrightarrow (E_1, \mathsf{broken})}\,(\textsc{SeqNormal})$$

The final environment is $\{x \mapsto 1\}$ and the outcome is broken. The assignment $x \leftarrow 2$ stands in no judgment of the tree: it does not run. The root is an instance of (SeqNormal) with $\omega = \mathsf{broken}$.

Exercise 4 Rules of a loop whose body ends normally

Given a loop while e do s, write two rules: the rule for a guard whose value is false, and the rule for a guard whose value is true and a body whose run ends with the outcome normal. Both follow (WhileFalse) and (WhileTrue) of block 2; what is to decide is the outcome in each premise and in each conclusion.

Solution.

$$\dfrac{E, e \twoheadrightarrow \mathsf{false}}{E, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow (E, \mathsf{normal})}\,(\textsc{WhileFalse})$$

$$\dfrac{E, e \twoheadrightarrow \mathsf{true} \quad E, s \twoheadrightarrow (E_1, \mathsf{normal}) \quad E_1, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow (E_2, \omega)}{E, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow (E_2, \omega)}\,(\textsc{WhileNormal})$$

In (WhileNormal) the outcome of the loop is the outcome ω of the rest of its run, the third premise. Exercise 14 shows that this ω is always normal; the rule need not say so.

Exercise 5 Rule of a loop whose body ends on a break

Given a loop while e do s whose body ends with the outcome broken, write the rule of the loop. The body runs only when the guard has the value true, and the rule has a premise that says so. The outcome of the loop itself is the point of the rule.

Solution.

$$\dfrac{E, e \twoheadrightarrow \mathsf{true} \quad E, s \twoheadrightarrow (E_1, \mathsf{broken})}{E, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow (E_1, \mathsf{normal})}\,(\textsc{WhileBreak})$$

The loop ends in the environment E_1 that the body leaves, and its outcome is normal: the loop *catches* the outcome broken, and the statement after the loop runs. The loop is the only construct that catches it; the sequence and the conditional pass it on.

Exercise 6 Rule of a loop whose body ends on a continue

Given a loop while e do s whose body ends with the outcome continued, write the rule of the loop. The rest of the body is skipped, and the loop itself goes on.

Solution.

$$\frac{E, e \twoheadrightarrow \mathsf{true} \quad E, s \twoheadrightarrow (E_1, \mathsf{continued}) \quad E_1, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow (E_2, \omega)}{E, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow (E_2, \omega)}\ (\textsc{WhileContinue})$$

The rule is (WhileNormal) with continued in place of normal in the second premise. When statements follow the continue in the body, (SeqContinue) has skipped them inside the run of s; the loop then goes on from E_1 with a new evaluation of the guard, exactly as after a body that ended normally.

Exercise 7 Runs of two loops

Given the environment $E_0 = \{x \mapsto 0,\ i \mapsto 0\}$ and the two statements

$$s_a = \mathsf{while}\ i < 2\ \mathsf{do}\ (i \leftarrow i + 1;\ \mathsf{if}\ i = 2\ \mathsf{then}\ \mathsf{break}\ \mathsf{else}\ x \leftarrow x + i)$$
$$s_b = \mathsf{while}\ i < 3\ \mathsf{do}\ (i \leftarrow i + 1;\ \mathsf{if}\ i = 2\ \mathsf{then}\ \mathsf{continue}\ \mathsf{else}\ x \leftarrow x + i)$$

write, for each of them, the run from E_0 as a list of steps, one line for each evaluation of the guard: the rule of the loop that applies and, when the body runs, the environment after the body and the outcome of the body. Write then the final environment and the outcome of each statement. The full derivation tree is not asked.

Solution. *The statement s_a.* The body is $i \leftarrow i + 1$; if $i = 2$ then break else $x \leftarrow x + i$.

1. In $\{x \mapsto 0,\ i \mapsto 0\}$ the guard $i < 2$ has the value true. The assignment gives $i \mapsto 1$; the guard $i = 2$ of the conditional has the value false, and $x \leftarrow x + i$ gives $x \mapsto 1$. The body ends in $\{x \mapsto 1,\ i \mapsto 1\}$ with the outcome normal: (WhileNormal).
2. In $\{x \mapsto 1,\ i \mapsto 1\}$ the guard has the value true. The assignment gives $i \mapsto 2$; the guard $i = 2$ has the value true, and break ends with broken by (Break), then (IfTrue). The body is a sequence whose first statement ends normally and whose second ends with broken: by (SeqNormal), with $\omega =$ broken, the body ends in $\{x \mapsto 1,\ i \mapsto 2\}$ with the outcome broken: (WhileBreak).

Final environment $\{x \mapsto 1,\ i \mapsto 2\}$, outcome normal. The rule (SeqBreak) is not used: it is the rule of a sequence whose *first* statement ends with broken.

The statement s_b. The body is $i \leftarrow i + 1$; if $i = 2$ then continue else $x \leftarrow x + i$.

1. In $\{x \mapsto 0,\ i \mapsto 0\}$ the guard $i < 3$ has the value true. The body ends in $\{x \mapsto 1,\ i \mapsto 1\}$ with the outcome normal: (WhileNormal).
2. In $\{x \mapsto 1,\ i \mapsto 1\}$ the guard has the value true. The assignment gives $i \mapsto 2$, and the conditional runs continue, which ends with continued. By (SeqNormal), with $\omega =$ continued, the body ends in $\{x \mapsto 1,\ i \mapsto 2\}$ with the outcome continued: (WhileContinue). The value of x is unchanged.
3. In $\{x \mapsto 1,\ i \mapsto 2\}$ the guard has the value true. The assignment gives $i \mapsto 3$, and $x \leftarrow x + i$ gives $x \mapsto 4$. The body ends in $\{x \mapsto 4,\ i \mapsto 3\}$ with the outcome normal: (WhileNormal).
4. In $\{x \mapsto 4,\ i \mapsto 3\}$ the guard $i < 3$ has the value false: (WhileFalse).

Final environment $\{x \mapsto 4,\ i \mapsto 3\}$, outcome normal.

Typing rules of the extension

A break or a continue that stands in no loop has no loop to end or to go on with. The type checker rejects such a program. The typing judgment for expressions, $\Gamma \vdash e : \tau$, is unchanged. The typing judgment for statements carries a *flag* ℓ:

$$(\Gamma, \ell) \vdash s \qquad\qquad \ell \in \{\mathsf{true}, \mathsf{false}\}$$

The flag says whether the statement s stands inside a loop. A whole program s is *well typed in* Γ when $(\Gamma, \mathsf{false}) \vdash s$.

Four typing rules are given. They are the rules of block 2, and each of them passes the flag from its conclusion to each premise on a statement, unchanged:

$$\frac{}{(\Gamma, \ell) \vdash \mathsf{skip}}\ (\text{T-Skip}) \qquad \frac{(\Gamma, \ell) \vdash s_1 \quad (\Gamma, \ell) \vdash s_2}{(\Gamma, \ell) \vdash s_1; s_2}\ (\text{T-Seq}) \qquad \frac{\Gamma \vdash x : \tau \quad \Gamma \vdash e : \tau}{(\Gamma, \ell) \vdash x \leftarrow e}\ (\text{T-Assign})$$

$$\frac{\Gamma \vdash e : \mathsf{bool} \quad (\Gamma, \ell) \vdash s_1 \quad (\Gamma, \ell) \vdash s_2}{(\Gamma, \ell) \vdash \mathsf{if}\ e\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2}\ (\text{T-If})$$

Exercise 8 Typing rules of break and continue

Given the judgment $(\Gamma, \ell) \vdash s$, write the typing rule of break and the typing rule of continue. Explain then how the rules reject break when the flag is false.

Solution.

$$\frac{}{(\Gamma, \mathsf{true}) \vdash \mathsf{break}}\ (\text{T-Break}) \qquad \frac{}{(\Gamma, \mathsf{true}) \vdash \mathsf{continue}}\ (\text{T-Continue})$$

The flag in each conclusion is the constant true, not a variable ℓ. No rule has a conclusion of the form $(\Gamma, \mathsf{false}) \vdash \mathsf{break}$, so that judgment has no derivation: the rules reject a program by the *absence* of a rule, as they reject $1 + \mathsf{true}$.

Exercise 9 Typing rule of the loop

Given a loop $\mathsf{while}\ e\ \mathsf{do}\ s$, write its typing rule. Write which flag the rule gives to the body s, and which flags it allows in its conclusion, with the reason for each.

Solution.

$$\frac{\Gamma \vdash e : \mathsf{bool} \quad (\Gamma, \mathsf{true}) \vdash s}{(\Gamma, \ell) \vdash \mathsf{while}\ e\ \mathsf{do}\ s}\ (\text{T-While})$$

The body is checked with the flag true, because every statement of the body stands inside this loop. The conclusion allows both flags: a loop is legal at the top of a program and inside another loop, and in both places its body stands inside a loop. The guard is an expression, and its judgment carries no flag.

Exercise 10 The flag in the rule of the conditional

Given the variant of (T-If) whose premises on s_1 and s_2 use the flag false in place of ℓ,

$$\frac{\Gamma \vdash e : \mathsf{bool} \quad (\Gamma, \mathsf{false}) \vdash s_1 \quad (\Gamma, \mathsf{false}) \vdash s_2}{(\Gamma, \ell) \vdash \mathsf{if}\ e\ \mathsf{then}\ s_1\ \mathsf{else}\ s_2},$$

write a program that the rules with (T-If) accept and that the rules with the variant reject, and explain which of the two rules is the right one.

Solution. The statement $\mathsf{while}\ \mathsf{true}\ \mathsf{do}\ (\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ \mathsf{break}\ \mathsf{else}\ \mathsf{skip})$ is such a program, in any Γ. With (T-If), the rule (T-While) checks the conditional with the flag true, (T-If) passes true to its branches, and $(\Gamma, \mathsf{true}) \vdash \mathsf{break}$ holds by (T-Break). With the variant, the branch is checked as $(\Gamma, \mathsf{false}) \vdash \mathsf{break}$, which has no derivation.

The rule (T-If) is the right one. A conditional is not a loop: a break in one of its branches ends the loop that contains the conditional, if there is one. Whether the branch stands inside a loop is therefore what it is for the conditional itself, and the flag passes unchanged. The rule (T-While) is the only rule whose premise may have another flag than its conclusion.

Exercise 11 Typing of a loop with a break

Given $\Gamma = \{x \mapsto \mathsf{int}\}$ and the statement

$$\mathsf{while}\ x > 0\ \mathsf{do}\ (\mathsf{if}\ x = 1\ \mathsf{then}\ \mathsf{break}\ \mathsf{else}\ x \leftarrow x - 1),$$

decide whether the statement is well typed in Γ. Write the derivation if there is one, and otherwise the premise that fails.

Solution. The statement is well typed in Γ. With $B = \mathsf{if}\ x = 1\ \mathsf{then}\ \mathsf{break}\ \mathsf{else}\ A$ and $A = x \leftarrow x - 1$:

$$\dfrac{\Gamma \vdash x > 0 : \mathsf{bool} \qquad \dfrac{\Gamma \vdash x = 1 : \mathsf{bool} \qquad \dfrac{}{(\Gamma, \mathsf{true}) \vdash \mathsf{break}}\ (\text{T-Break}) \qquad \dfrac{\Gamma \vdash x : \mathsf{int} \qquad \Gamma \vdash x - 1 : \mathsf{int}}{(\Gamma, \mathsf{true}) \vdash A}\ (\text{T-Assign})}{(\Gamma, \mathsf{true}) \vdash B}\ (\text{T-If})}{(\Gamma, \mathsf{false}) \vdash \mathsf{while}\ x > 0\ \mathsf{do}\ B}\ (\text{T-While})$$

The derivations of the four judgments on expressions at the leaves are not drawn; they hold by (T-Var) and (T-Int), with the rules of $>$, $=$ and $-$; for instance $\Gamma \vdash x > 0 : \mathsf{bool}$ has the premises $\Gamma \vdash x : \mathsf{int}$ and $\Gamma \vdash 0 : \mathsf{int}$. The flag is false at the root and true from the body of the loop upward.

Exercise 12 A conditional outside and inside a loop

Given $\Gamma = \{x \mapsto \mathsf{int}\}$ and the statement $P = \mathsf{if}\ x > 0\ \mathsf{then}\ \mathsf{break}\ \mathsf{else}\ \mathsf{skip}$, decide

(a) whether $(\Gamma, \mathsf{false}) \vdash P$ is derivable;
(b) whether $(\Gamma, \mathsf{false}) \vdash \mathsf{while}\ \mathsf{true}\ \mathsf{do}\ P$ is derivable.

Write for each the rules that apply and, where the derivation fails, the premise that fails.

Solution. **(a)** Not derivable. The only rule for a conditional is (T-If); with the flag false in its conclusion, its second premise is $(\Gamma, \mathsf{false}) \vdash \mathsf{break}$. The only rule for break is (T-Break), whose conclusion has the flag true. The premise has no derivation.

(b) Derivable. (T-While) has the premises $\Gamma \vdash \mathsf{true} : \mathsf{bool}$, which holds by (T-Bool), and $(\Gamma, \mathsf{true}) \vdash P$. (T-If) has the premises $\Gamma \vdash x > 0 : \mathsf{bool}$, which holds, $(\Gamma, \mathsf{true}) \vdash \mathsf{break}$, which holds by (T-Break), and $(\Gamma, \mathsf{true}) \vdash \mathsf{skip}$, which holds by (T-Skip).

The same statement P is rejected alone and accepted as the body of a loop: the flag makes the typing of a statement depend on the place where it stands.

Exercise 13 Nested loops

Given $\Gamma = \{i \mapsto \mathsf{int},\ j \mapsto \mathsf{int}\}$ and the statement

$$\mathsf{while}\ i < 10\ \mathsf{do}\ \big((\mathsf{while}\ j < 5\ \mathsf{do}\ \mathsf{break});\ i \leftarrow i + 1\big),$$

(a) decide whether the statement is well typed in Γ, and write the flag with which each of its sub-statements is checked;
(b) write which loop the break ends in a run, with the big-step rules that say so;
(c) decide whether the typing judgment tells which loop a break ends.

Solution. **(a)** Well typed. The outer loop is checked with the flag false. Its body, the sequence, is checked with true by (T-While), and so are the inner loop and $i \leftarrow i + 1$, by (T-Seq). The body of the inner loop, break, is checked with true by (T-While), and (T-Break) applies. The guards $i < 10$ and $j < 5$ have the type bool, and $\Gamma \vdash i : \mathsf{int}$, $\Gamma \vdash i + 1 : \mathsf{int}$.

(b) The inner loop. When the guard $j < 5$ has the value true, the body break ends with broken

by (Break), and (WhileBreak) gives the *inner* loop the outcome normal. The sequence then runs $i \leftarrow i + 1$ by (SeqNormal), and the outer loop goes on by (WhileNormal): the outcome broken never reaches it.

(c) No. The flag has two values: it says that a break stands inside *some* loop, not inside which one, nor inside how many. Both loops check their body with the same flag true. Which loop a break ends is fixed by the semantics, and it is the innermost loop that contains it.

The property that the typing guarantees

The semantics gives a result to a break that stands in no loop: the run ends with the outcome broken. The typing rules exist to exclude that result. The claim is

$$\text{if } (\Gamma, \mathsf{false}) \vdash s \text{ and } E, s \twoheadrightarrow (E', \omega), \text{ then } \omega = \mathsf{normal}. \qquad (*)$$

Exercise 14 Outcome of a well-typed program

Given the big-step rules and the typing rules of the extension, and the claim $(*)$,

(a) decide whether the claim still holds with the flag true in place of false; write a counterexample if it does not;
(b) write a claim about $(\Gamma, \ell) \vdash s$ that holds for both values of ℓ and that gives $(*)$ when $\ell = \mathsf{false}$;
(c) prove that claim by induction on the derivation of $E, s \twoheadrightarrow (E', \omega)$, one case for each big-step rule. The case of the loop first needs a property of the loop alone: the outcomes with which the run of a loop can end.

Solution. **(a)** The claim fails with the flag true. The statement break is a counterexample: $(\Gamma, \mathsf{true}) \vdash \mathsf{break}$ holds by (T-Break), and $E, \mathsf{break} \twoheadrightarrow (E, \mathsf{broken})$ by (Break).

(b) For every Γ, ℓ, s, E, E' and ω:

$$\text{if } (\Gamma, \ell) \vdash s \text{ and } E, s \twoheadrightarrow (E', \omega), \text{ then } \omega = \mathsf{normal} \text{ or } \ell = \mathsf{true}. \qquad (**)$$

With $\ell = \mathsf{false}$ the second alternative is excluded, which gives $(*)$.

(c) *A property of the loop alone.* Every derivation of $E, \mathsf{while}\ e\ \mathsf{do}\ s \twoheadrightarrow (E', \omega)$ has $\omega = \mathsf{normal}$. The proof is by induction on the derivation. It ends with one of the four rules of the loop. (WhileFalse) and (WhileBreak) have normal in their conclusion. In (WhileNormal) and (WhileContinue) the outcome of the conclusion is the outcome of the last premise, which is a smaller derivation for the same loop: by the induction hypothesis that outcome is normal.

Proof of $(**)$, by induction on the derivation of $E, s \twoheadrightarrow (E', \omega)$. Each statement form has exactly one typing rule, so the derivation of $(\Gamma, \ell) \vdash s$ ends with that rule, and its premises hold.

- (Skip), (Assign): $\omega = \mathsf{normal}$.
- (Break), (Continue): the statement is break or continue, typed by (T-Break) or (T-Continue), whose conclusion has $\ell = \mathsf{true}$.
- (SeqNormal): ω is the outcome of the run of s_2. (T-Seq) gives $(\Gamma, \ell) \vdash s_2$, and the induction hypothesis on the run of s_2 gives $\omega = \mathsf{normal}$ or $\ell = \mathsf{true}$.
- (SeqBreak), (SeqContinue): the run of s_1 ends with an outcome that is not normal. (T-Seq) gives $(\Gamma, \ell) \vdash s_1$, and the induction hypothesis on the run of s_1 leaves $\ell = \mathsf{true}$.
- (IfTrue), (IfFalse): ω is the outcome of the run of a branch. (T-If) gives that branch the same flag ℓ, and the induction hypothesis on its run concludes.
- The four rules of the loop: $\omega = \mathsf{normal}$, by the property of the loop.

The typing environment Γ plays no part in the proof: the property is about the flag alone. The claim says nothing of a run that does not end, since it has no derivation.