CHAPTER

The Triggering of Events

4

So far, we have addressed that when an event occurs, a fluent changes its truth value. What about the opposite? That is, when a fluent changes its truth value, an event occurs; more generally, when a particular condition becomes true, an event occurs. We call such an event a *triggered event*. An example of a triggered event is a ball bouncing off of a wall when it reaches the wall. This chapter addresses the triggering of events in response to conditions; we also discuss triggered fluents.

4.1 TRIGGER AXIOMS

We specify when a triggered event occurs using a trigger axiom.

Definition 4.1. If γ is a condition, α is an event term, and τ is a timepoint term, then

$$\gamma \Rightarrow \mathit{Happens}(\alpha, \tau)$$

is a ***trigger axiom***.

4.1.1 EXAMPLE: ALARM CLOCK

Whenever we use an alarm clock, we perform commonsense reasoning about triggered events. The alarm going off is a triggered event, and we can formalize the operation of an alarm clock using trigger axioms.

We start with some effect axioms. If a clock's alarm time is t_1 and an agent sets the clock's alarm time to t_2, then the clock's alarm time will be t_2 and will no longer be t_1:

$$\begin{gathered}\mathit{HoldsAt}(\mathit{AlarmTime}(c, t_1), t) \land t_1 \neq t_2 \Rightarrow \\ \mathit{Initiates}(\mathit{SetAlarmTime}(a, c, t_2), \mathit{AlarmTime}(c, t_2), t)\end{gathered} \tag{4.1}$$

$$\begin{gathered}\mathit{HoldsAt}(\mathit{AlarmTime}(c, t_1), t) \land t_1 \neq t_2 \Rightarrow \\ \mathit{Terminates}(\mathit{SetAlarmTime}(a, c, t_2), \mathit{AlarmTime}(c, t_1), t)\end{gathered} \tag{4.2}$$

If an agent turns on a clock's alarm, then it will be on:

$$\mathit{Initiates}(\mathit{TurnOnAlarm}(a, c), \mathit{AlarmOn}(c), t) \tag{4.3}$$

If an agent turns off a clock's alarm, then it will no longer be on:

$$Terminates(TurnOffAlarm(a, c), AlarmOn(c), t) \tag{4.4}$$

If an alarm starts beeping, then it will be beeping:

$$Initiates(StartBeeping(c), Beeping(c), t) \tag{4.5}$$

If an agent turns off a clock's alarm, then the clock will no longer be beeping:

$$Terminates(TurnOffAlarm(a, c), Beeping(c), t) \tag{4.6}$$

We have a state constraint that says that a clock has a unique alarm time at any given time:

$$\begin{gathered} HoldsAt(AlarmTime(c, t_1), t) \wedge HoldsAt(AlarmTime(c, t_2), t) \Rightarrow \\ t_1 = t_2 \end{gathered} \tag{4.7}$$

Now we use a trigger axiom. If a clock's alarm time is the present moment and the alarm is on, then the clock starts beeping:

$$\begin{gathered} HoldsAt(AlarmTime(c, t), t) \wedge HoldsAt(AlarmOn(c), t) \Rightarrow \\ Happens(StartBeeping(c), t) \end{gathered} \tag{4.8}$$

Let us use the following observations and narrative. At timepoint 0, the alarm is not on, the alarm is not beeping, and the alarm time is set to 10:

$$\neg HoldsAt(AlarmOn(Clock), 0) \tag{4.9}$$

$$\neg HoldsAt(Beeping(Clock), 0) \tag{4.10}$$

$$HoldsAt(AlarmTime(Clock, 10), 0) \tag{4.11}$$

$$\neg ReleasedAt(f, t) \tag{4.12}$$

At timepoint 0, Nathan sets the alarm clock for timepoint 2; and at timepoint 1, he turns on the alarm:

$$Happens(SetAlarmTime(Nathan, Clock, 2), 0) \tag{4.13}$$

$$Happens(TurnOnAlarm(Nathan, Clock), 1) \tag{4.14}$$

We can then show that the alarm clock will be beeping at timepoint 3.

Proposition 4.1. *Let* $\Sigma = (4.1) \wedge (4.2) \wedge (4.3) \wedge (4.4) \wedge (4.5) \wedge (4.6)$, $\Delta = (4.8) \wedge (4.13) \wedge (4.14)$, $\Omega = U[SetAlarmTime, TurnOnAlarm, TurnOffAlarm, StartBeeping] \wedge U[AlarmTime, AlarmOn, Beeping]$, $\Psi = (4.7)$, *and* $\Gamma = (4.9) \wedge (4.10) \wedge (4.11) \wedge (4.12)$. *Then we have*

$$\begin{gathered} CIRC[\Sigma; Initiates, Terminates, Releases] \wedge CIRC[\Delta; Happens] \wedge \\ \Omega \wedge \Psi \wedge \Gamma \wedge \mathrm{DEC} \vdash HoldsAt(Beeping(Clock), 3) \end{gathered}$$

Proof. From $CIRC[\Sigma; Initiates, Terminates, Releases]$ and Theorems 2.1 and 2.2, we have

$$\begin{aligned} & Initiates(e, f, t) \Leftrightarrow \\ & \exists a, c, t_1, t_2\, (e = SetAlarmTime(a, c, t_2)\ \wedge \\ & f = AlarmTime(c, t_2)\ \wedge \end{aligned} \tag{4.15}$$

$$HoldsAt(AlarmTime(c, t_1), t) \wedge$$
$$t_1 \neq t_2) \vee$$
$$\exists a, c\, (e = TurnOnAlarm(a, c) \wedge f = AlarmOn(c)) \vee$$
$$\exists c\, (e = StartBeeping(c) \wedge f = Beeping(c)) \tag{4.16}$$

$$Terminates(e, f, t) \Leftrightarrow$$
$$\exists a, c, t_1, t_2\, (e = SetAlarmTime(a, c, t_2) \wedge$$
$$f = AlarmTime(c, t_1) \wedge$$
$$HoldsAt(AlarmTime(c, t_1), t) \wedge$$
$$t_1 \neq t_2) \vee$$
$$\exists a, c\, (e = TurnOffAlarm(a, c) \wedge f = AlarmOn(c)) \vee$$
$$\exists a, c\, (e = TurnOffAlarm(a, c) \wedge f = Beeping(c))$$

$$\neg Releases(e, f, t) \tag{4.17}$$

From $CIRC[\Delta; Happens]$ and Theorem 2.1, we have

$$Happens(e, t) \Leftrightarrow \tag{4.18}$$
$$\exists c\ (e = StartBeeping(c) \wedge$$
$$HoldsAt(AlarmTime(c, t), t) \wedge$$
$$HoldsAt(AlarmOn(c), t)) \vee$$
$$(e = SetAlarmTime(Nathan, Clock, 2) \wedge t = 0) \vee$$
$$(e = TurnOnAlarm(Nathan, Clock) \wedge t = 1)$$

From (4.13) (which follows from (4.18)), (4.11), $10 \neq 2$, (4.1) (which follows from (4.15)), and DEC9, we have

$$HoldsAt(AlarmTime(Clock, 2), 1) \tag{4.19}$$

From (4.18) and (4.16), we have $\neg\exists e\, (Happens(e, 1) \wedge Terminates(e, AlarmTime(Clock, 2), 1))$. From this, (4.19), (4.12), and DEC5, we have

$$HoldsAt(AlarmTime(Clock, 2), 2) \tag{4.20}$$

From (4.14) (which follows from (4.18)), (4.3) (which follows from (4.15)), and DEC9, we have

$$HoldsAt(AlarmOn(Clock), 2) \tag{4.21}$$

From (4.20), (4.21), and (4.8) (which follows from (4.18)), we have $Happens(StartBeeping(Clock), 2)$. From this, (4.5) (which follows from (4.15)), and DEC9, we have $HoldsAt(Beeping(Clock), 3)$. ■

4.2 PREVENTING REPEATED TRIGGERING

As long as the condition of a trigger axiom is true, an event is triggered. We must take steps to ensure that events are not repeatedly triggered. In the example in Section 4.1.1, the trigger axiom (4.8) does not repeatedly trigger because the condition is true only at a single instant, when the alarm timepoint equals the current timepoint. In most other cases, we will require a method to prevent repeated triggering.

This may be accomplished using trigger axioms and effect axioms of the form

$$\neg HoldsAt(\beta, \tau) \wedge \gamma \Rightarrow Happens(\alpha, \tau)$$

$$Initiates(\alpha, \beta, \tau)$$

or

$$HoldsAt(\beta, \tau) \wedge \gamma \Rightarrow Happens(\alpha, \tau)$$

$$Terminates(\alpha, \beta, \tau).$$

4.2.1 EXAMPLE: BANK ACCOUNT SERVICE FEE

An example in which we prevent the repeated triggering of events in this way is the charging of service fees. We formalize a bank account that charges a monthly service fee if the balance falls below a minimum amount at any time during the month.

First, we describe the basic operation of the account using effect axioms. If the balance of account a_1 is greater than or equal to some amount and that amount is transferred from a_1 to account a_2, then the balance of a_2 increases by that amount and the balance of a_1 decreases by that amount:

$$\begin{gathered} HoldsAt(Balance(a_1, x_1), t) \wedge \\ HoldsAt(Balance(a_2, x_2), t) \wedge \\ x_3 > 0 \wedge x_1 \geq x_3 \Rightarrow \\ Initiates(Transfer(a_1, a_2, x_3), Balance(a_2, x_2 + x_3), t) \end{gathered} \tag{4.22}$$

$$\begin{gathered} HoldsAt(Balance(a_1, x_1), t) \wedge \\ HoldsAt(Balance(a_2, x_2), t) \wedge \\ x_3 > 0 \wedge x_1 \geq x_3 \Rightarrow \\ Terminates(Transfer(a_1, a_2, x_3), Balance(a_2, x_2), t) \end{gathered} \tag{4.23}$$

$$\begin{gathered} HoldsAt(Balance(a_1, x_1), t) \wedge \\ HoldsAt(Balance(a_2, x_2), t) \wedge \\ x_3 > 0 \wedge x_1 \geq x_3 \Rightarrow \\ Initiates(Transfer(a_1, a_2, x_3), Balance(a_1, x_1 - x_3), t) \end{gathered} \tag{4.24}$$

$$\begin{gathered} HoldsAt(Balance(a_1, x_1), t) \wedge \\ HoldsAt(Balance(a_2, x_2), t) \wedge \end{gathered} \tag{4.25}$$

$$x_3 > 0 \land x_1 \geq x_3 \Rightarrow$$
$$Terminates(Transfer(a_1, a_2, x_3), Balance(a_1, x_1), t)$$

We have a state constraint that says that an account has a unique balance at any given time:

$$HoldsAt(Balance(a, x_1), t) \land HoldsAt(Balance(a, x_2), t) \Rightarrow x_1 = x_2 \tag{4.26}$$

We also have a trigger axiom that says that if the balance in an account falls below the minimum balance and a service fee has not yet been charged to the account, then a service fee is charged to the account:

$$\begin{aligned} &HoldsAt(Balance(a, x), t) \land \\ &x < MinimumBalance(a) \land \\ &\neg HoldsAt(ServiceFeeCharged(a), t) \Rightarrow \\ &Happens(ChargeServiceFee(a), t) \end{aligned} \tag{4.27}$$

When a service fee is charged to an account, a note is made of this fact so that the account is not repeatedly charged:

$$Initiates(ChargeServiceFee(a), ServiceFeeCharged(a), t) \tag{4.28}$$

This is reset once each month:

$$EndOfMonth(t) \Rightarrow Happens(MonthlyReset(a), t) \tag{4.29}$$
$$Terminates(MonthlyReset(a), ServiceFeeCharged(a), t) \tag{4.30}$$

If a service fee is charged to an account, then the balance of the account decreases by the amount of the service fee:

$$\begin{aligned} &HoldsAt(Balance(a, x), t) \Rightarrow \\ &Initiates(ChargeServiceFee(a), Balance(a, x - ServiceFee(a)), t) \end{aligned} \tag{4.31}$$

$$\begin{aligned} &HoldsAt(Balance(a, x), t) \Rightarrow \\ &Terminates(ChargeServiceFee(a), Balance(a, x), t) \end{aligned} \tag{4.32}$$

Let us use the following observations and narrative about two bank accounts. Initially, a service fee has not been charged to the first account, the balance in both accounts is 1000, the minimum balance of the first account is 500, and the service fee of the first account is 5:

$$\neg HoldsAt(ServiceFeeCharged(Account1), 0) \tag{4.33}$$
$$HoldsAt(Balance(Account1, 1000), 0) \tag{4.34}$$
$$HoldsAt(Balance(Account2, 1000), 0) \tag{4.35}$$
$$MinimumBalance(Account1) = 500 \tag{4.36}$$
$$ServiceFee(Account1) = 5 \tag{4.37}$$
$$\neg ReleasedAt(f, t) \tag{4.38}$$

Two transfers are made from the first account to the second account. A transfer of 200 is made and then a transfer of 400 is made:

$$Happens(Transfer(Account1, Account2, 200), 0) \tag{4.39}$$
$$Happens(Transfer(Account1, Account2, 400), 1) \tag{4.40}$$

We can show that, after these transfers, the balance in the first account will be 395.

Proposition 4.2. *Let* $\Sigma = (4.22) \wedge (4.23) \wedge (4.24) \wedge (4.25) \wedge (4.28) \wedge (4.30) \wedge (4.31) \wedge (4.32)$, $\Delta = (4.27) \wedge (4.29) \wedge (4.39) \wedge (4.40)$, $\Omega = U[Transfer, ChargeServiceFee, MonthlyReset] \wedge U[Balance, ServiceFeeCharged]$, $\Psi = (4.26)$, *and* $\Gamma = (4.33) \wedge (4.34) \wedge (4.35) \wedge (4.36) \wedge (4.37) \wedge (4.38)$. *Then we have*

$$CIRC[\Sigma; Initiates, Terminates, Releases] \wedge CIRC[\Delta; Happens] \wedge$$
$$\Omega \wedge \Psi \wedge \Gamma \wedge \text{DEC} \vdash HoldsAt(Balance(Account1, 395), 3)$$

Proof. From $CIRC[\Sigma; Initiates, Terminates, Releases]$ and Theorems 2.1 and 2.2, we have

$$\begin{aligned}
&Initiates(e,f,t) \Leftrightarrow \\
&\exists a_1, a_2, x_1, x_2, x_3\ (e = Transfer(a_1, a_2, x_3) \wedge \\
&f = Balance(a_2, x_2 + x_3) \wedge \\
&HoldsAt(Balance(a_1, x_1), t) \wedge \\
&HoldsAt(Balance(a_2, x_2), t) \wedge \\
&x_3 > 0 \wedge x_1 \geq x_3) \vee \\
&\exists a_1, a_2, x_1, x_2, x_3\ (e = Transfer(a_1, a_2, x_3) \wedge \\
&f = Balance(a_1, x_1 - x_3) \wedge \\
&HoldsAt(Balance(a_1, x_1), t) \wedge \\
&HoldsAt(Balance(a_2, x_2), t) \wedge \\
&x_3 > 0 \wedge x_1 \geq x_3) \vee \\
&\exists a\ (e = ChargeServiceFee(a) \wedge f = ServiceFeeCharged(a)) \vee \\
&\exists a, x\ (e = ChargeServiceFee(a) \wedge \\
&f = Balance(a, x - ServiceFee(a)) \wedge \\
&HoldsAt(Balance(a, x), t))
\end{aligned} \tag{4.41}$$

$$\begin{aligned}
&Terminates(e,f,t) \Leftrightarrow \\
&\exists a_1, a_2, x_1, x_2, x_3\ (e = Transfer(a_1, a_2, x_3) \wedge \\
&f = Balance(a_2, x_2) \wedge \\
&HoldsAt(Balance(a_1, x_1), t) \wedge \\
&HoldsAt(Balance(a_2, x_2), t) \wedge \\
&x_3 > 0 \wedge x_1 \geq x_3) \vee \\
\\
&\exists a_1, a_2, x_1, x_2, x_3\ (e = Transfer(a_1, a_2, x_3) \wedge \\
&f = Balance(a_1, x_1) \wedge \\
&HoldsAt(Balance(a_1, x_1), t) \wedge \\
&HoldsAt(Balance(a_2, x_2), t) \wedge \\
&x_3 > 0 \wedge x_1 \geq x_3) \vee \\
&\exists a\ (e = MonthlyReset(a) \wedge f = ServiceFeeCharged(a)) \vee
\end{aligned} \tag{4.42}$$

$$\begin{aligned}
&\exists a,x\,(e = ChargeServiceFee(a) \wedge \\
&f = Balance(a,x) \wedge \\
&HoldsAt(Balance(a,x),t)) \\
&\neg Releases(e,f,t)
\end{aligned} \tag{4.43}$$

From $CIRC[\Delta; Happens]$ and Theorem 2.1, we have

$$\begin{aligned}
&Happens(e,t) \Leftrightarrow \\
&\exists a,x\,(e = ChargeServiceFee(a) \wedge \\
&HoldsAt(Balance(a,x),t) \wedge \\
&x < MinimumBalance(a) \wedge \\
&\neg HoldsAt(ServiceFeeCharged(a),t)) \vee \\
&\exists a\,(e = MonthlyReset(a) \wedge EndOfMonth(t)) \vee \\
&(e = Transfer(Account1, Account2, 200) \wedge t = 0) \vee \\
&(e = Transfer(Account1, Account2, 400) \wedge t = 1)
\end{aligned} \tag{4.44}$$

From (4.39) (which follows from (4.44)), (4.34), (4.35), $200 > 0$, $1000 \geq 200$, (4.24) (which follows from (4.41)), and DEC9, we have

$$HoldsAt(Balance(Account1, 800), 1) \tag{4.45}$$

From (4.39) (which follows from (4.44)), (4.34), (4.35), $200 > 0$, $1000 \geq 200$, (4.22) (which follows from (4.41)), and DEC9, we have

$$HoldsAt(Balance(Account2, 1200), 1) \tag{4.46}$$

From (4.26), (4.34), (4.36), $\neg(1000 < 500)$, and (4.44), we have $\neg Happens(ChargeServiceFee(Account1), 0)$. From this, (4.44), and (4.41), we have $\neg\exists e\,(Happens(e,0) \wedge Initiates(e, ServiceFeeCharged(Account1), 0))$. From this, (4.33), (4.38), and DEC6, we have

$$\neg HoldsAt(ServiceFeeCharged(Account1), 1) \tag{4.47}$$

From (4.40) (which follows from (4.44)), (4.45), (4.46), $400 > 0$, $800 \geq 400$, (4.24) (which follows from (4.41)), and DEC9, we have

$$HoldsAt(Balance(Account1, 400), 2) \tag{4.48}$$

From (4.26), (4.45), (4.36), $\neg(800 < 500)$, and (4.44), we have $\neg Happens(ChargeServiceFee(Account1), 1)$. From this, (4.44), and (4.41), we have $\neg\exists e\,(Happens(e,1) \wedge Initiates(e, ServiceFeeCharged(Account1), 1))$. From this, (4.47), (4.38), and DEC6, we have

$$\neg HoldsAt(ServiceFeeCharged(Account1), 2)$$

From this, (4.48), (4.36), $400 < 500$, and (4.27) (which follows from (4.44)), we have $Happens(ChargeServiceFee(Account1), 2)$. From this, (4.48), (4.37), (4.31) (which follows from (4.41)), and DEC9, we have $HoldsAt(Balance(Account1, 395), 3)$. ∎

4.3 TRIGGERED FLUENTS

So far, we have discussed how a trigger axiom is used to represent that a certain event occurs when a certain condition becomes true. What if we would like to represent that a fluent becomes true (or false) when a condition becomes true? This cannot be represented directly in the event calculus. Instead, we must introduce an event that is triggered by the condition and that initiates or terminates the fluent.

Thus, we represent that the condition γ initiates a fluent β as follows:

$$\gamma \Rightarrow Happens(\alpha, \tau)$$
$$Initiates(\alpha, \beta, t)$$

We represent that the condition γ terminates a fluent β as follows:

$$\gamma \Rightarrow Happens(\alpha, \tau)$$
$$Terminates(\alpha, \beta, t)$$

BIBLIOGRAPHIC NOTES

Shanahan (1990) introduced an early form of the trigger axiom into a simplified version of the original event calculus (Kowalski & Sergot, 1986). Shanahan (1995a, pp. 268-272; 1997b, pp. 305-313) uses a predicate $Triggers(s, e)$ to represent that an event e occurs in state s. Trigger axioms in the form used in this book were introduced by Shanahan (1996, p. 685) and are discussed in detail by Shanahan (1997b, pp. 258-265, 325-329). The method for representing triggered fluents is from Morgenstern (2001, p. 353).

Proposals for incorporating triggered events into the situation calculus have been made by Pinto (1994), R. Miller (1996), and Reiter (1996). See the discussion in the Bibliographic notes of Chapter 16. Pinto (1998a) uses triggered events in the situation calculus to represent the starting and stopping of current in an electrical circuit.

Tran and Baral (2004b) incorporate triggered events into an action language inspired by $\mathcal{A}$ (Gelfond & Lifschitz, 1993), implement the language as an answer set program (Baral, 2003), and apply triggered events to the modeling of molecular interactions in cells. The triggering rule (Tran & Baral, 2004b, p. 555)

$g_1, \ldots, g_m$ **n_triggers** b

represents that action b normally occurs when conditions $g_1, \ldots, g_m$ are true. The inhibition rule (p. 555)

$h_1, \ldots, h_l$ **inhibits** c

represents that action c does not occur when conditions $h_1, \ldots, h_l$ are true.

Triggered events are represented in action language $\mathcal{C}+$ E. Giunchiglia, Lee, Lifschitz, McCain, and Turner (2004) using action dynamic laws of the form (p. 70)

caused F **if** G

where F is an action and G is a condition. $\mathcal{C}+$ is discussed in Section 16.3.1 .

EXERCISES

4.1 Add a snooze alarm to the alarm clock axiomatization in Section 4.1.1. If, when the alarm is beeping, an agent presses the snooze button, then the alarm stops beeping and starts beeping again after nine timepoints.

4.2 Use the extended alarm clock axiomatization in Exercise 4.1 to prove that, if a particular alarm clock is set, the clock starts beeping at the appropriate time, and an agent hits the snooze button, then the alarm stops beeping after that time and is beeping 10 timepoints later.

4.3 Rework the axiomatization in Section 4.2.1 so that a monthly fee is charged at the end of each month rather than immediately.

4.4 Formalize the operation of a mousetrap. Prove that a mouse entering the trap is caught.

4.5 Formalize a price notification service for traders of financial instruments. When the price of a financial instrument falls below or rises above a certain level, the service informs the trader. Prove that, if the trader requests notification of when stock XYZ falls below 100, then when the stock falls below that level the service informs the trader.

4.6 Formalize that you introduce yourself when meeting someone for the first time.