Business Processes Modelling

MPB (6 cfu, 295AA)

Roberto Bruni
http://www.di.unipi.it/~bruni

23 - Quantitative Analysis

Object

We overview some techniques for the quantitative analysis of business processes

Ch.7 of Fundamental of Business Process Management. M. Dumas et al.
(inspired by slides available at https://courses.cs.ut.ee/2014/bpm/)

Performance Analysis

Validation

is concerned with the relation between the model and reality

Verification

is typically used to answer **qualitative** questions

Is there a deadlock possible? It is possible to successfully handle a specific case? Will all cases terminate eventually? It is possible to execute two tasks in any order?

Performance analysis

is typically used to answer **quantitative** questions

How many cases can be handled in one hour? What is the average flow time? How many extra resources are required? How many cases are handled within 2 days?

Performance dimensions

Any company would like to make its processes

faster,

cheaper,

and better.

Performance dimensions

Any company would like to make its processes

faster, (time)

cheaper,

and better.

Performance dimensions

Any company would like to make its processes

faster, (time)

cheaper, (finance)

and better.

Performance dimensions

Any company would like to make its processes

faster, (time)

cheaper, (finance)

and better. (quality)

KPI

To estimate the performance along any dimension
we need to fix something that can be measured

A **process performance measure** is
a quantity that can be unambiguously determined
for a given business process

Time, finance, quality
can be refined to a number of
Key Performance Indicators (KPI)

1. Time

Cycle time:
the time needed to handle one case from start to end.

Processing time (also **service time**):
the time that resources spend on actually handling the case

Waiting time:
the time that a case spends in *idle* mode
(e.g., it includes **queueing time** due to unavailability of resources to handle the case)

Time objectives

One can aim to reduce the *average* cycle time

or to reduce the *maximal* cycle time

or to *meet a cycle time* negotiated with the customer

2. Finance

Cost, turnover, yield or revenue are all concerned with finance-related performance dimensions.

A yield increase may have the same effect as a cost decrease w.r.t. the organization profit.

Business process redesign is typically concerned with cost.

Aggregation functions

There are several types of cost:
cost of production, of delivery, of human resources, …

Each type can be refined into performance measures by selecting an **aggregation function** such as:
count, average, variance, minimum, maximum, …

Example: average delivery cost per item

Cost types

Fixed cost:

overhead costs not affected by the intensity of processing (e.g., use of infrastructure, maintenance costs).

Variable cost:

positively correlated with some variable quantity (e.g. the level of sales, the number of purchased goods, the number of new hires)

Operational cost:

closer to productivity, often directly related to the output of a business process (e.g. labor cost in producing a good or delivering a service)

Operational cost

Process redesign is often aimed to **reduce operational cost**, particularly labor cost

Although task automation may reduce labor cost, it may cause incidental cost involved with developing the respective application and fixed maintenance cost for it

3. Quality

External quality: from the viewpoint of the client
(e.g. client satisfaction with the delivered product or with the way the process has been executed)
Important factors: amount, relevance, quality and timeliness of the information a client receives as process progresses

Internal quality: from the viewpoint of process participants
Important factors: the level of control of the work performed, of variation experienced, of challenges faced

Quality vs time

External process quality is often measured in terms of time (e.g. the average cycle time or the percentage of cases where deadlines are missed)

In the following we assume that any performance measure where time is involved is classified under the time dimension, even if it is related to quality

Deriving performance measures

One possible method for deriving performance measures for a given process is the following:

1. Formulate performance objectives of the process at the high level, in the form of a desirable state that the process should ideally reach

2. For each performance objective, identify the relevant performance dimensions and aggregation functions and derive one or more KPI for the objective

3. Define a target objective for each KPI

Deriving performance measures: example

A restaurant has recently lost many customers due to **poor customer service**.

The management team has decided to address this issue first of all by **focussing on the delivery of meals**.

The team gathered data by asking customers about how quickly they liked to receive their meals and what they considered as an **acceptable wait**.

The data suggested that half of the customers would prefer their meals to be served in **15 minutes or less**.

All customers agreed that a waiting time of **30 minutes or more** is unacceptable.

Deriving performance measures: example

1. Formulate performance objectives of the process at the high level, in the form of a desirable state that the process should ideally reach (e.g., customers should be served in less than 30 minutes)

2. For each performance objective, identify the relevant performance dimensions and aggregation functions and derive one or more KPI for the objective (e.g., time dimension, ST_{30} be the percentage of customers served in less than 30 minutes)

3. Define a target objective for each KPI (e.g., $ST_{30} \geq 97\%$)

Deriving performance measures: example

1. Formulate performance objectives of the process at the high level, in the form of a desirable state that the process should ideally reach (e.g., customers should be served in about 15 minutes)

2. For each performance objective, identify the relevant performance dimensions and aggregation functions and derive one or more KPI for the objective (e.g., time dimension, ST_{15} be the percentage of customers served in less than 15 minutes)

3. Define a target objective for each KPI (e.g., $ST_{15} \geq 85\%$)

Deriving performance measures: example

1. Formulate performance objectives of the process at the high level, in the form of a desirable state that the process should ideally reach (e.g., customers should be served in about 15 minutes)

2. For each performance objective, identify the relevant performance dimensions and aggregation functions and derive one or more KPI for the objective (e.g., time dimension, AMDT be the average meal delivery time)

3. Define a target objective for each KPI (e.g., AMDT ≤ 15’)

Typical process performance measures

Time

- Cycle time
- Waiting time / time spent in non-value-added tasks

Cost

- Cost per execution
- Resource utilization

Quality

- Error rates (negative outcomes, wrong info)
- Missed promise

Flow analysis

Flow analysis is a family of techniques
to estimate the overall performance of a process
given some knowledge about the performance of its activities

Examples:
we calculate the min/max/average **cycle time** of a process
given the min/max/average cycle time of each activity

we compute the average **cost** of a process
knowing the cost-per-execution of each activity.

we calculate the **error rate** of a process
given the error rate of each activity.

Cycle time analysis

Cycle time analysis

Cycle time = difference between
the start time (ready to be executed) and
the end time (completion) of a case

Cycle time analysis = the task of calculating the average cycle time of an entire process or some process fragment

Assumption: average activity times are available for all the activities involved in the process

Activity time = waiting time + processing time

Flow patterns

The simplest case is that of a single activity, but then we can take into account different structure patterns that frequently occur:

paths composed in sequence

alternative paths (XOR split and join)

parallel paths (AND split and join)

rework (1-or-more cycles, 0-or-more cycles)

Notation

We denote the **average cycle time** by **CT**
and call it simply *cycle time*

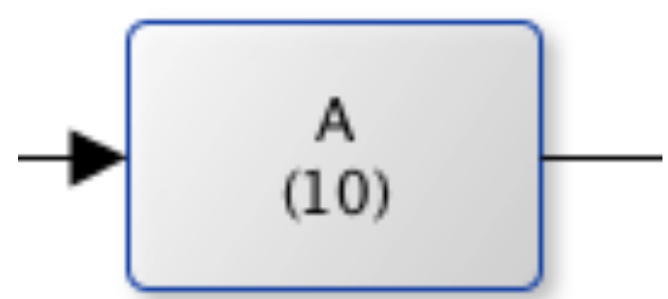
When several (sub)processes P_1, P_2, …, P_n are involved
we refer to their cycle times by CT_1, CT_2, …, CT_n

Similarly, if activities A, B, … are involved
we refer to their cycle times by CT_A, CT_B, …

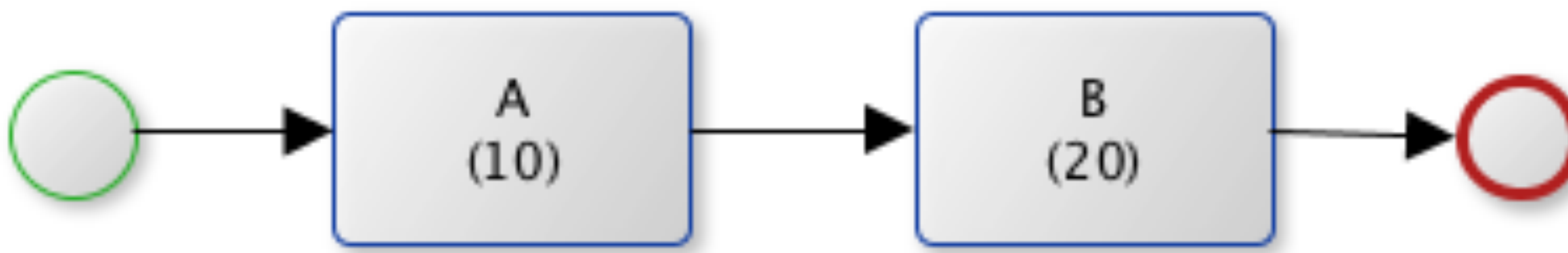
Notation

In diagrams, we will often write activity cycle time within parentheses

CT_A = 10 units of time

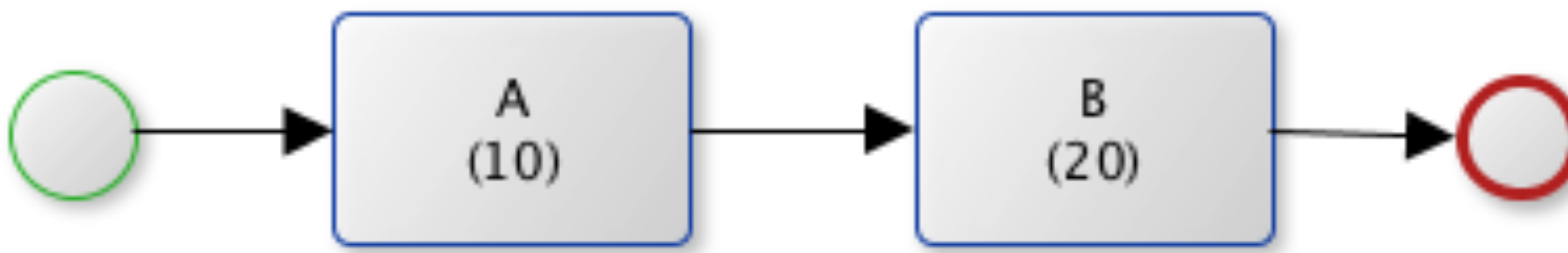


Sequence



CT = ?

Sequence



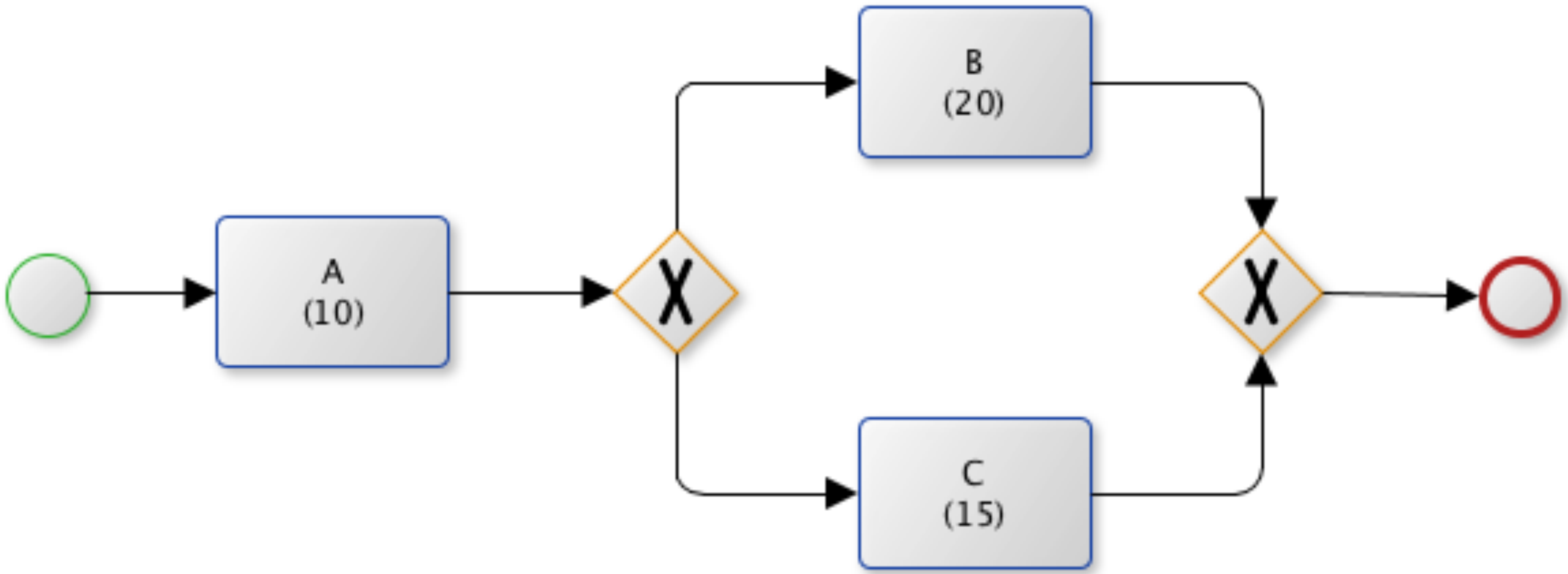
$$CT = CT_A + CT_B = 10 + 20 = 30$$

Sequence

The cycle time of a purely sequential fragment of a process is the sum of the cycle times of the activities in the fragment

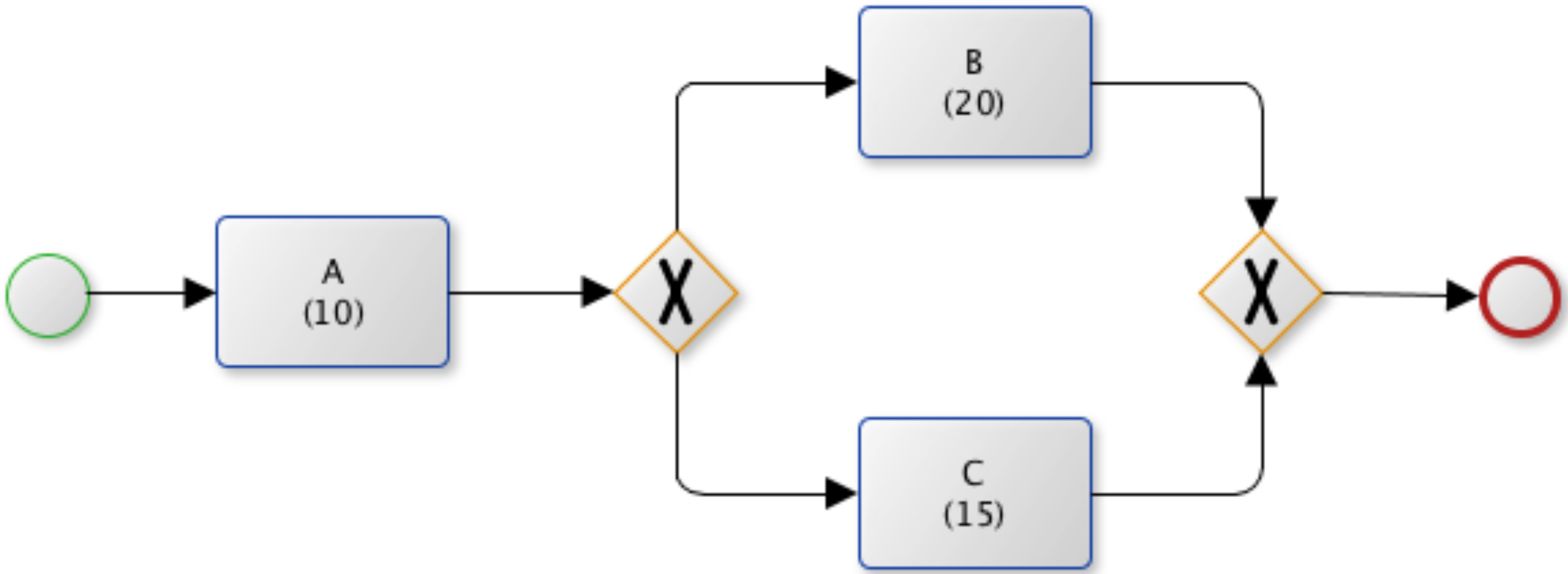
$$CT = \sum_{i=1}^{n} CT_i$$

Alternative paths



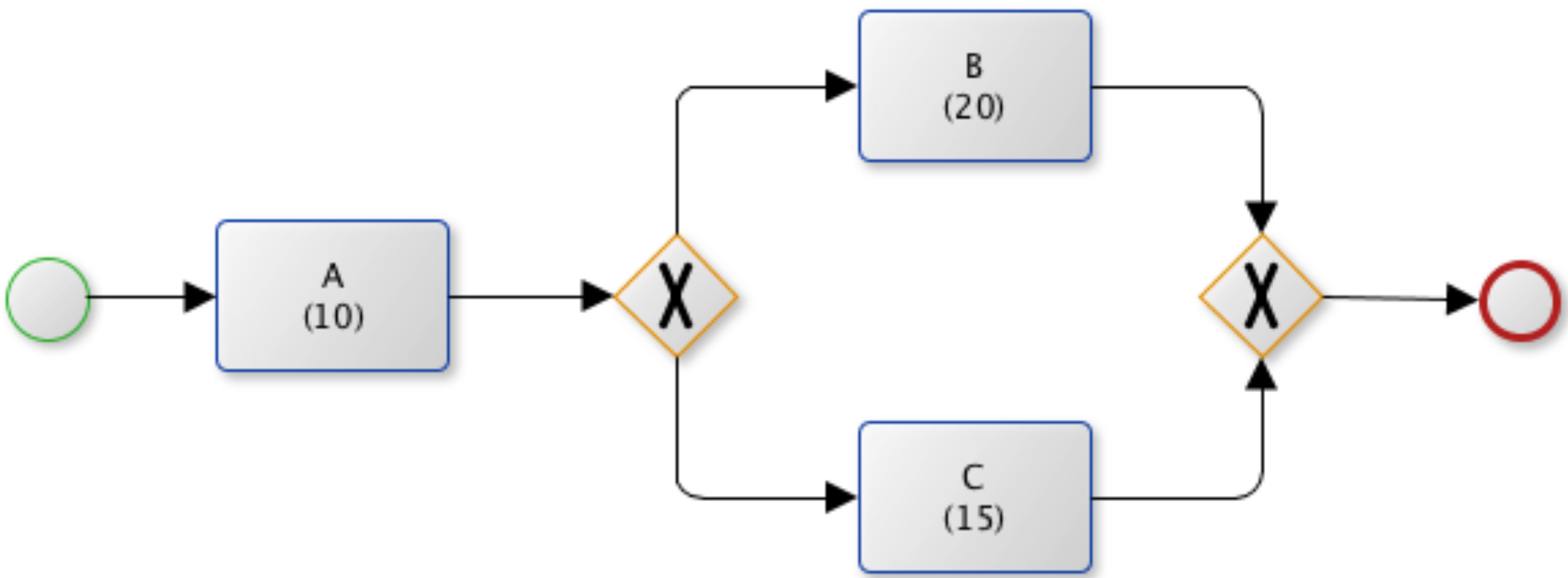
CT = ?

Alternative paths



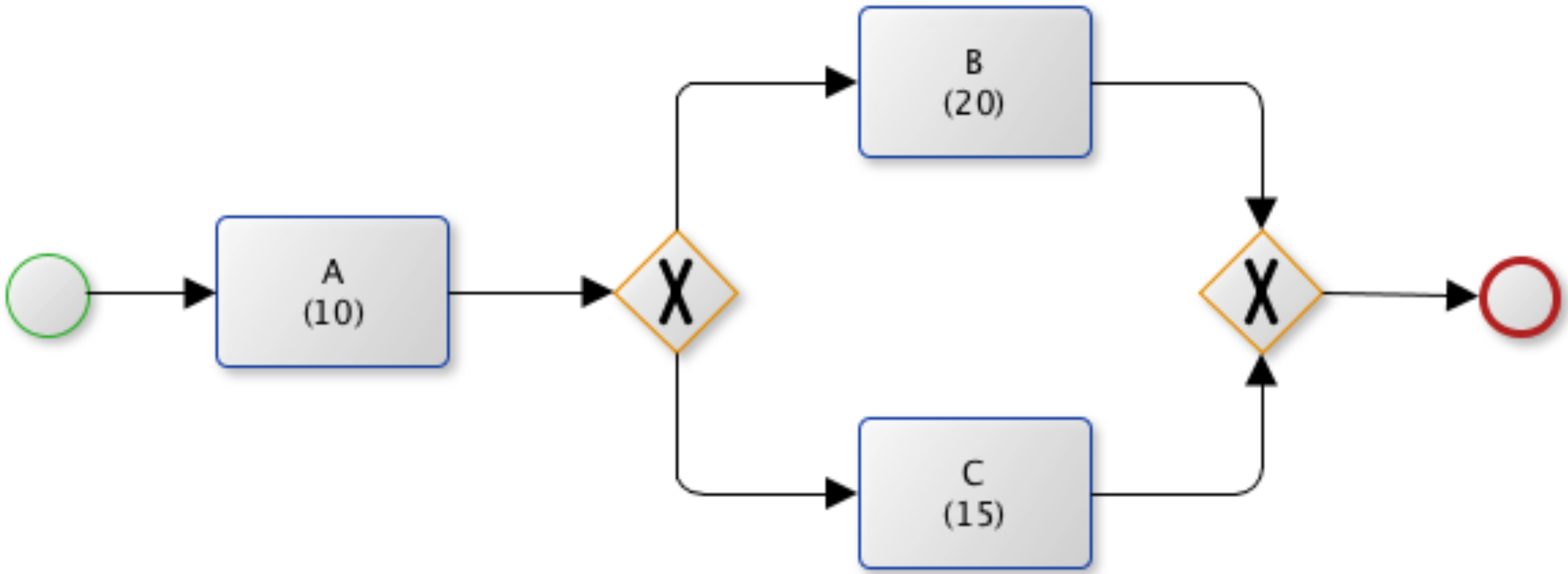
in some cases CT = $CT_A + CT_B = 10 + 20 =$ **30**
in other cases CT = $CT_A + CT_C = 10 + 15 =$ **25**
whether the average is closer to 25 or to 30
depends on **how frequently** each branch is taken

Alternative paths



if B and C are taken in 50% of the cases each, then the average sits in the middle between 25 and 30

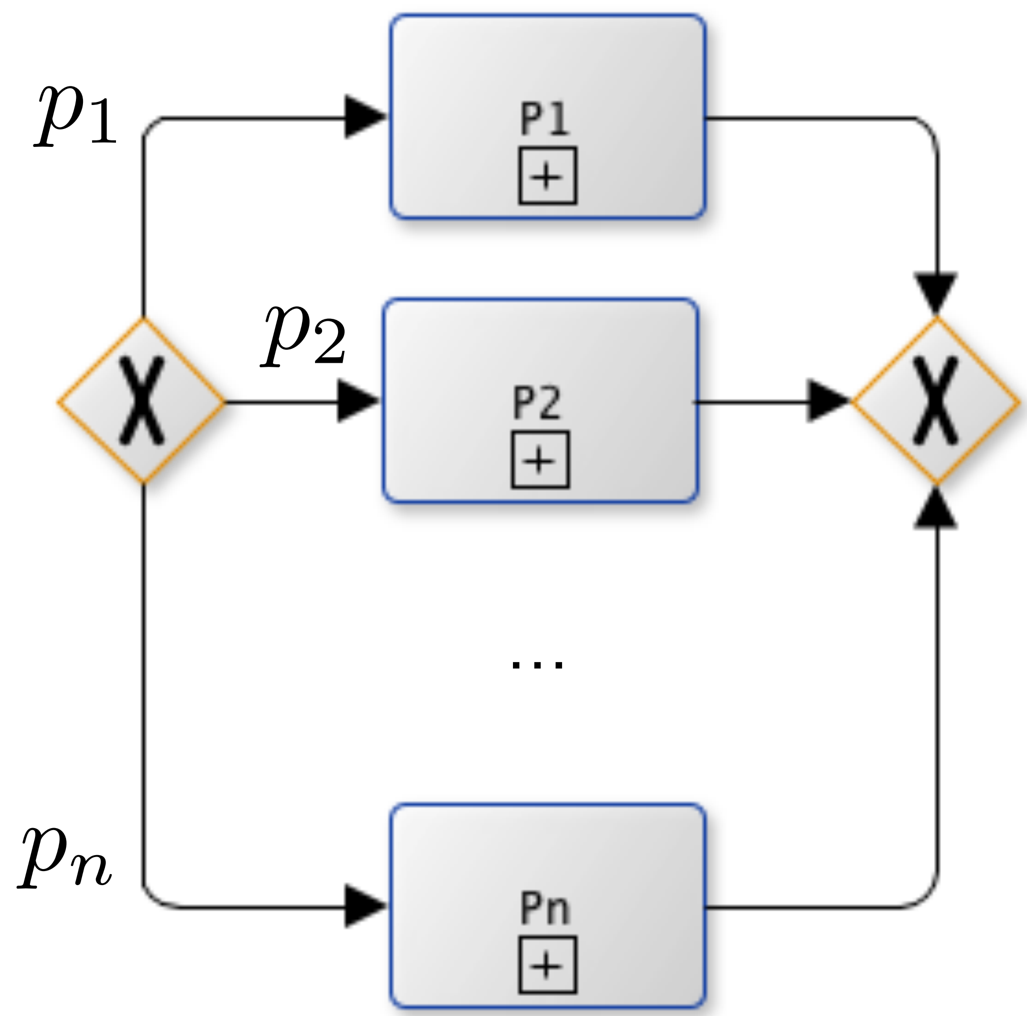
Alternative paths



if B is taken in 90% of the cases, then the average is closer to 30 than 25

Branching probability

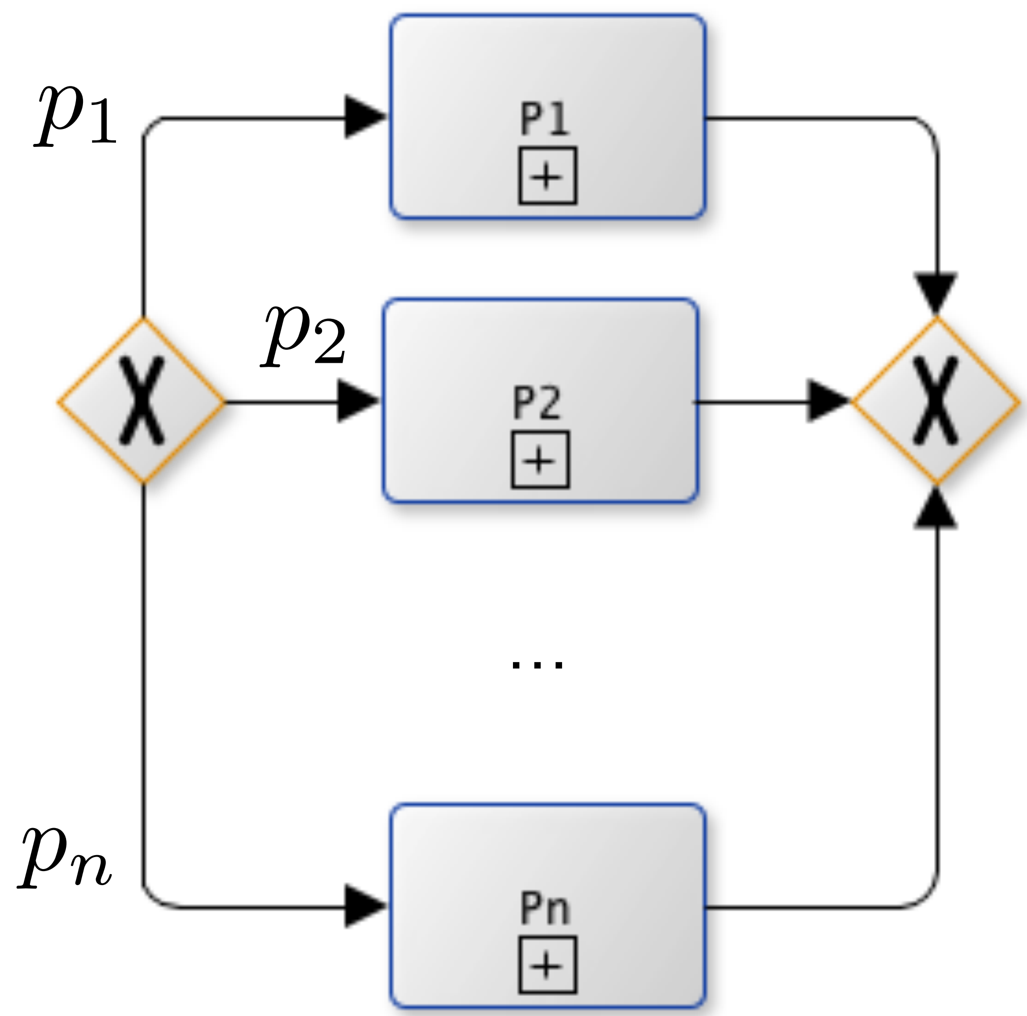
$$\sum_{i=1}^{n} p_i = 1$$



Branching probability p_i:
is the frequency with
which a given branch of a decision gateway is taken

Alternative paths

$$\sum_{i=1}^{n} p_i = 1$$

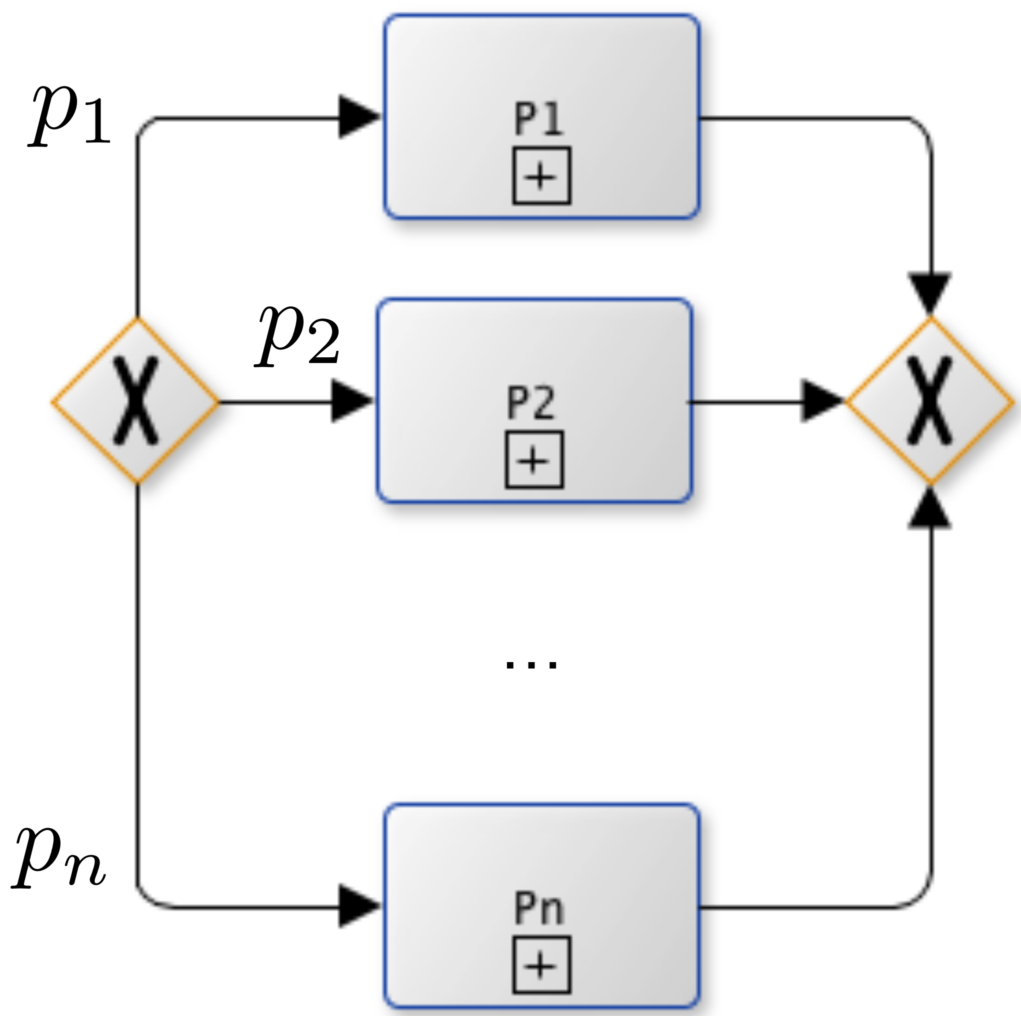


The fragment between the XOR split and join is called **XOR-block**

The cycle time of a XOR-block fragment is the **weighted average** of the cycle times of the branches

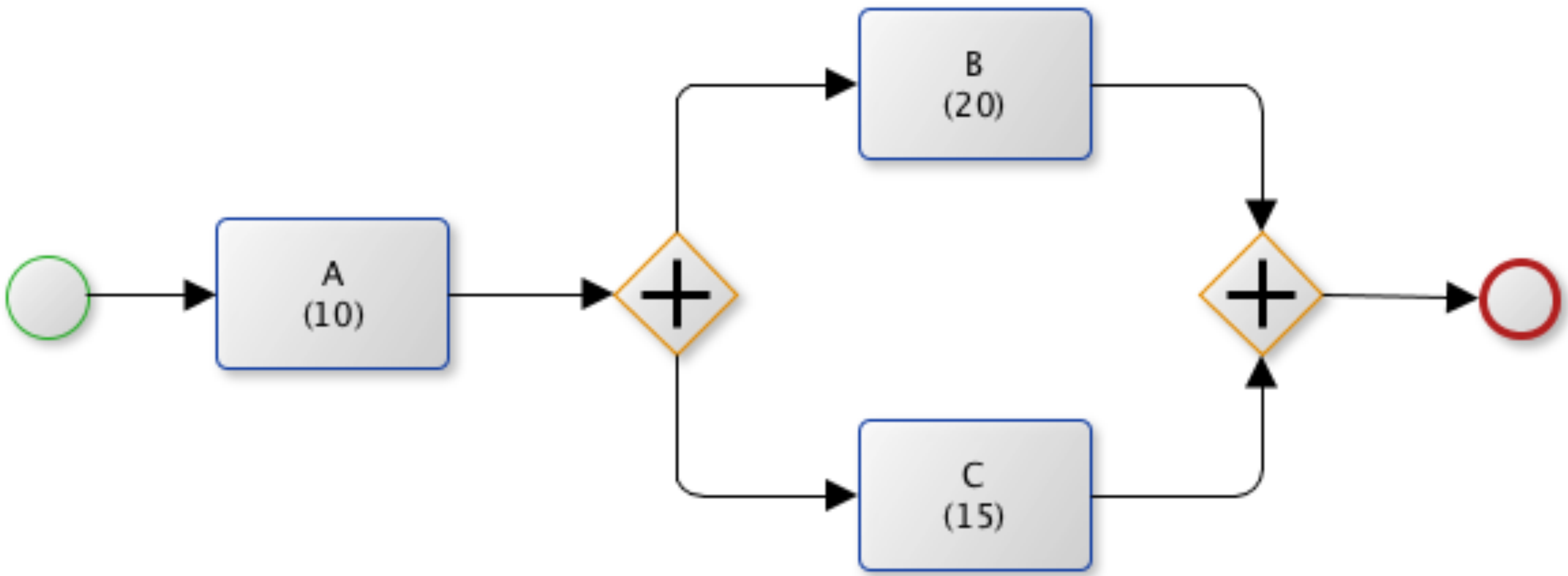
Alternative paths

$$\sum_{i=1}^{n} p_i = 1$$



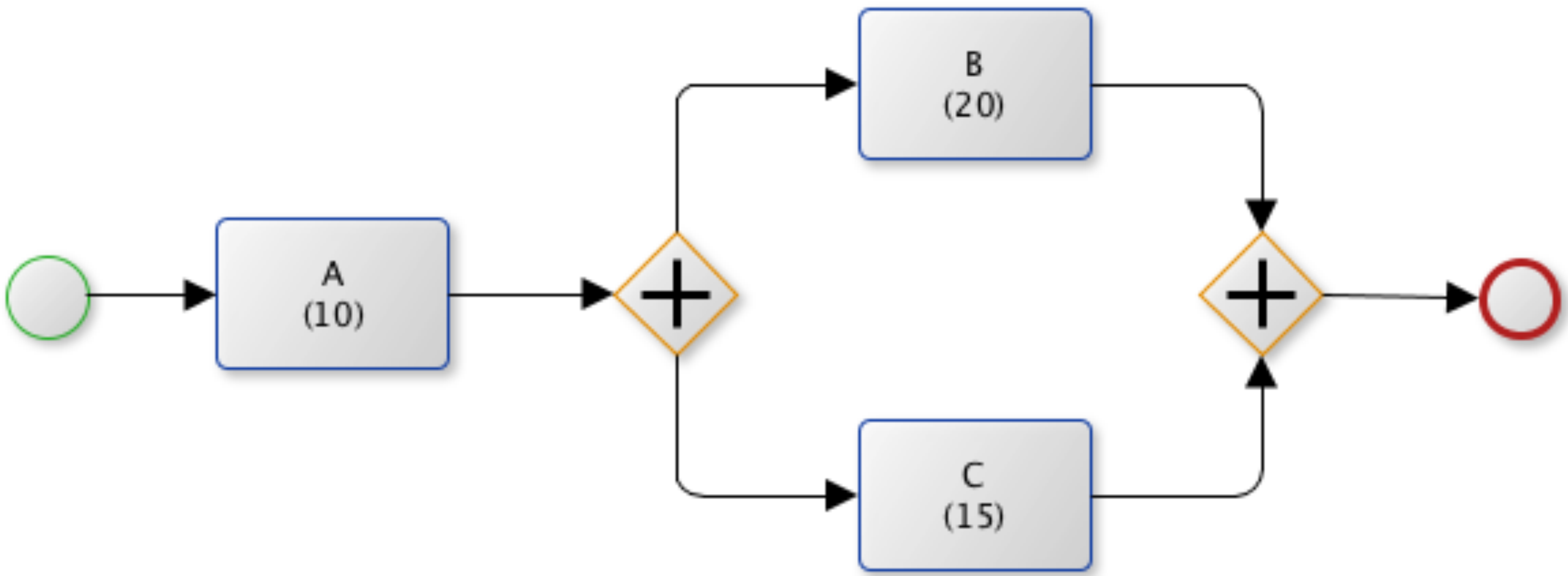
$$CT = \sum_{i=1}^{n} p_i \cdot CT_i$$

Parallel paths



$CT = CT_A + CT_B + CT_C = 10 + 20 + 15 = 45$?

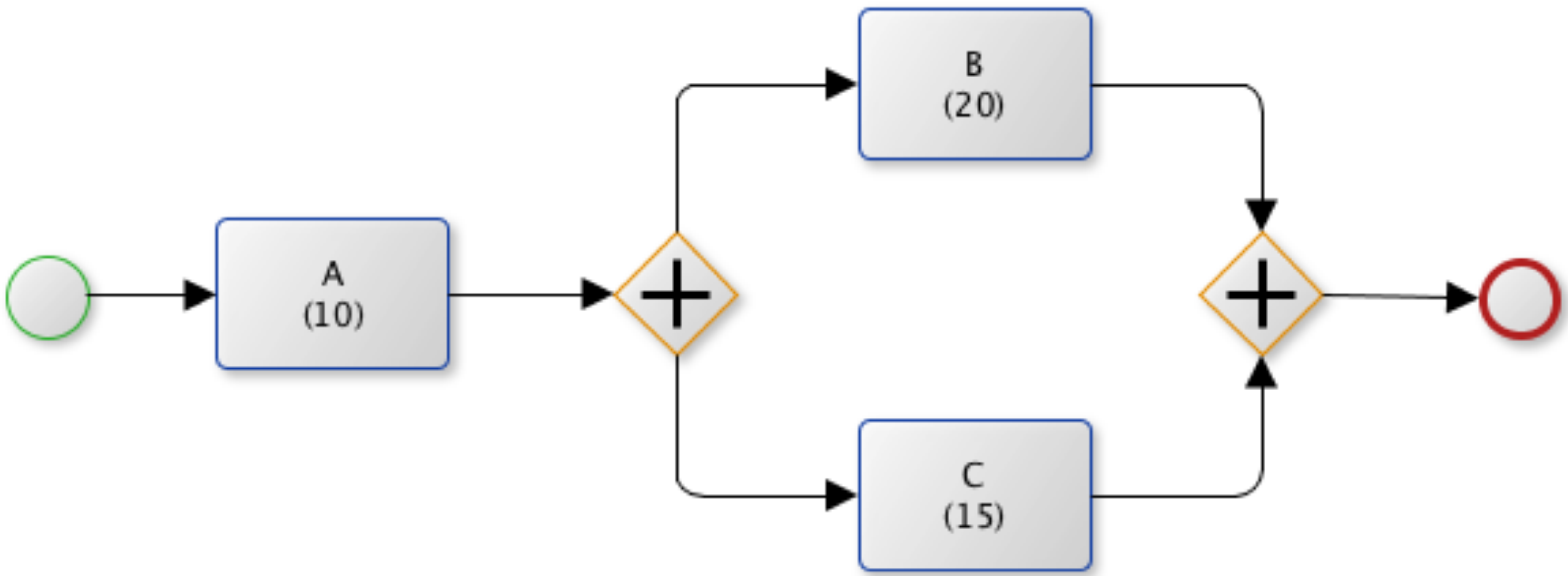
Parallel paths



$CT = CT_A + CT_B + CT_C = 10 + 20 + 15 = 45$?

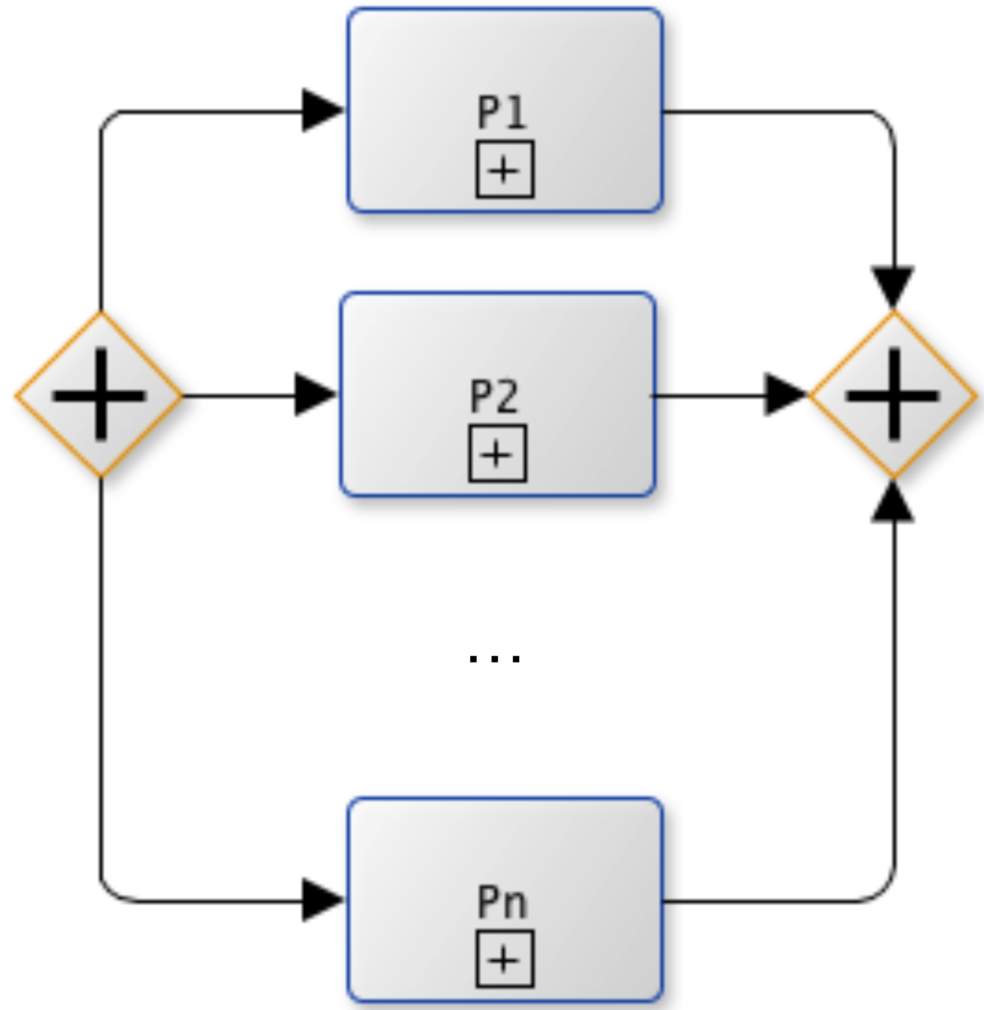
(but while B is executed, also C is executed, and B takes longer than C)

Parallel paths



CT = 10 + max {20,15} = 10 + 20 = 30

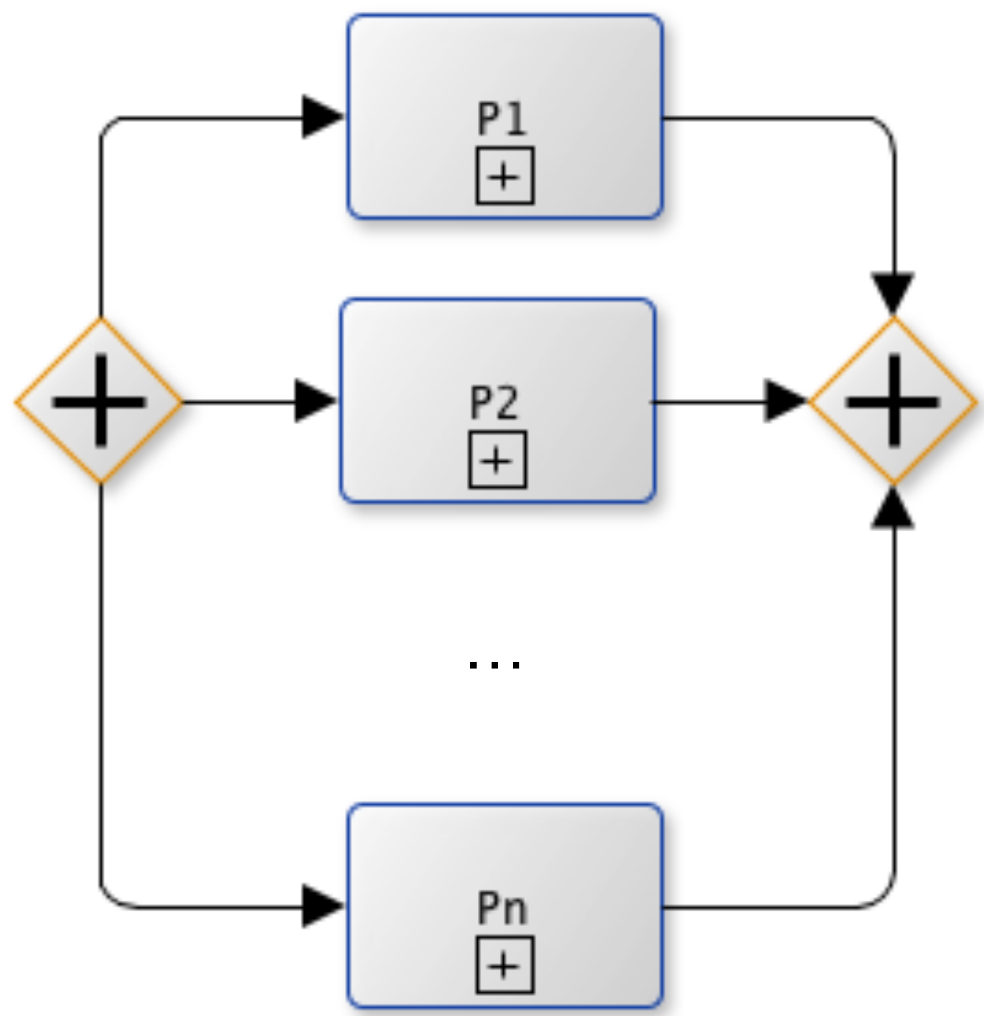
Parallel paths



The fragment between the AND split and join is called **AND-block**

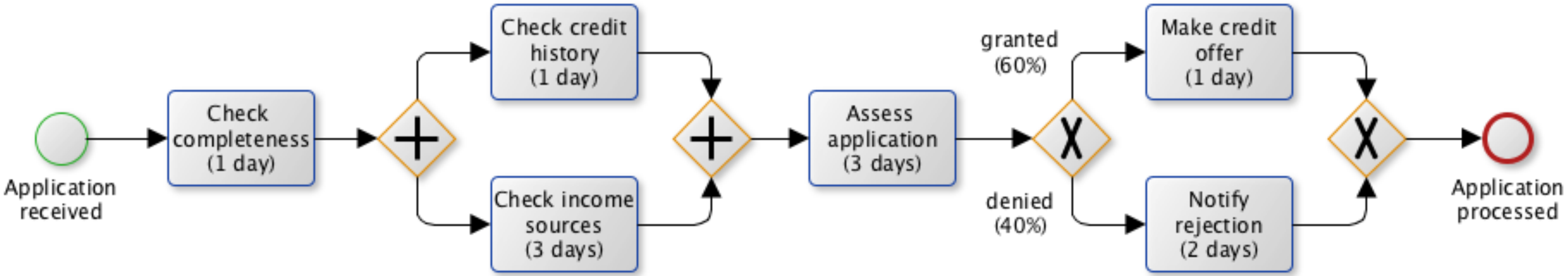
The cycle time of an AND-block fragment is the cycle time of the **slowest** branch

Parallel paths



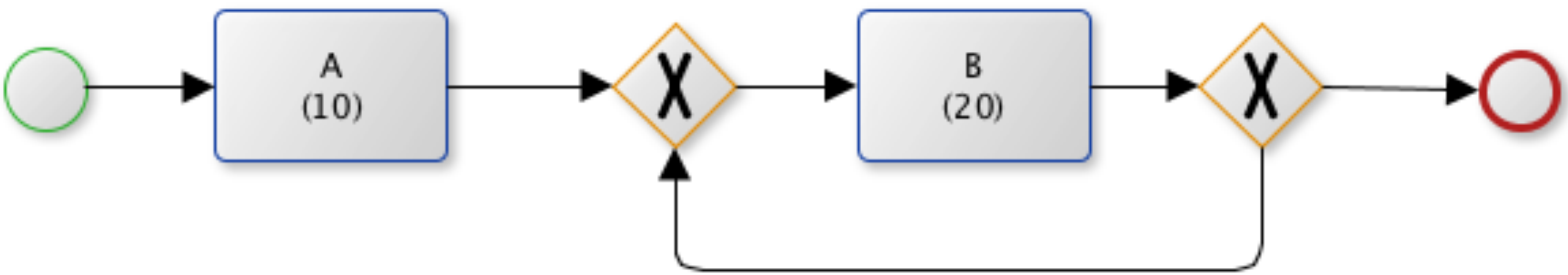
$$CT = \max_i\{CT_i\} = \max\{CT_1, CT_2, ..., CT_n\}$$

Question time



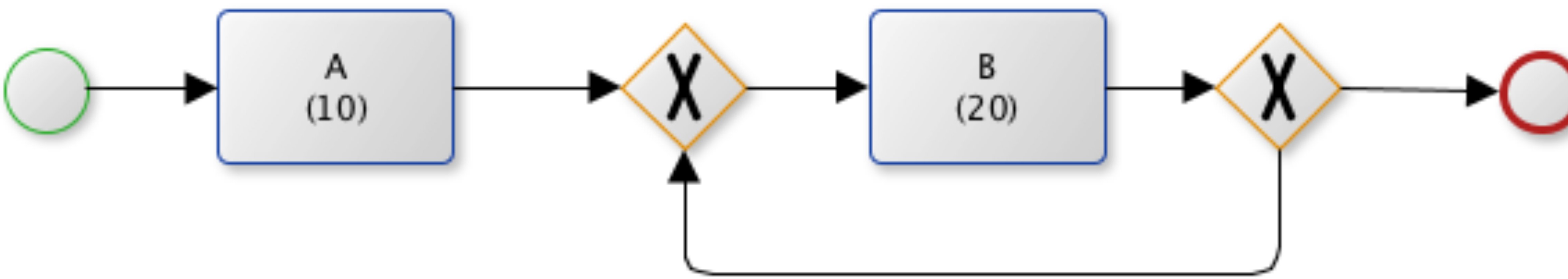
CT = ?

Rework loop
(1 or more times)



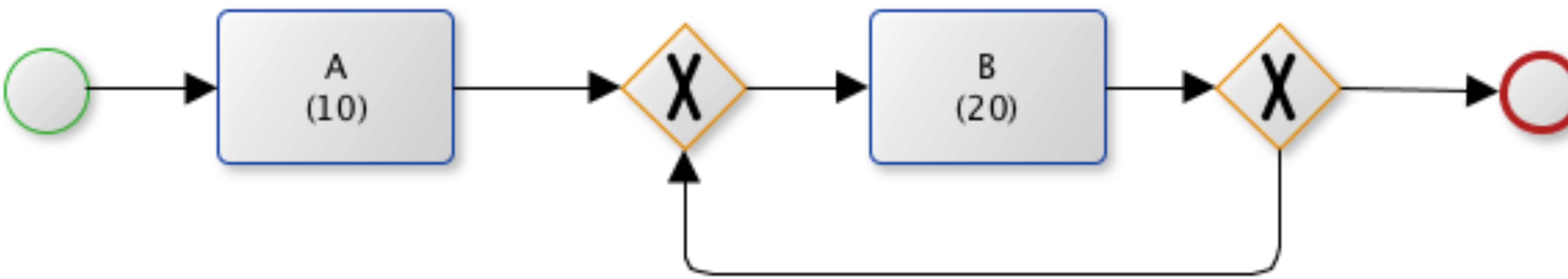
CT = ?

Rework loop
(1 or more times)



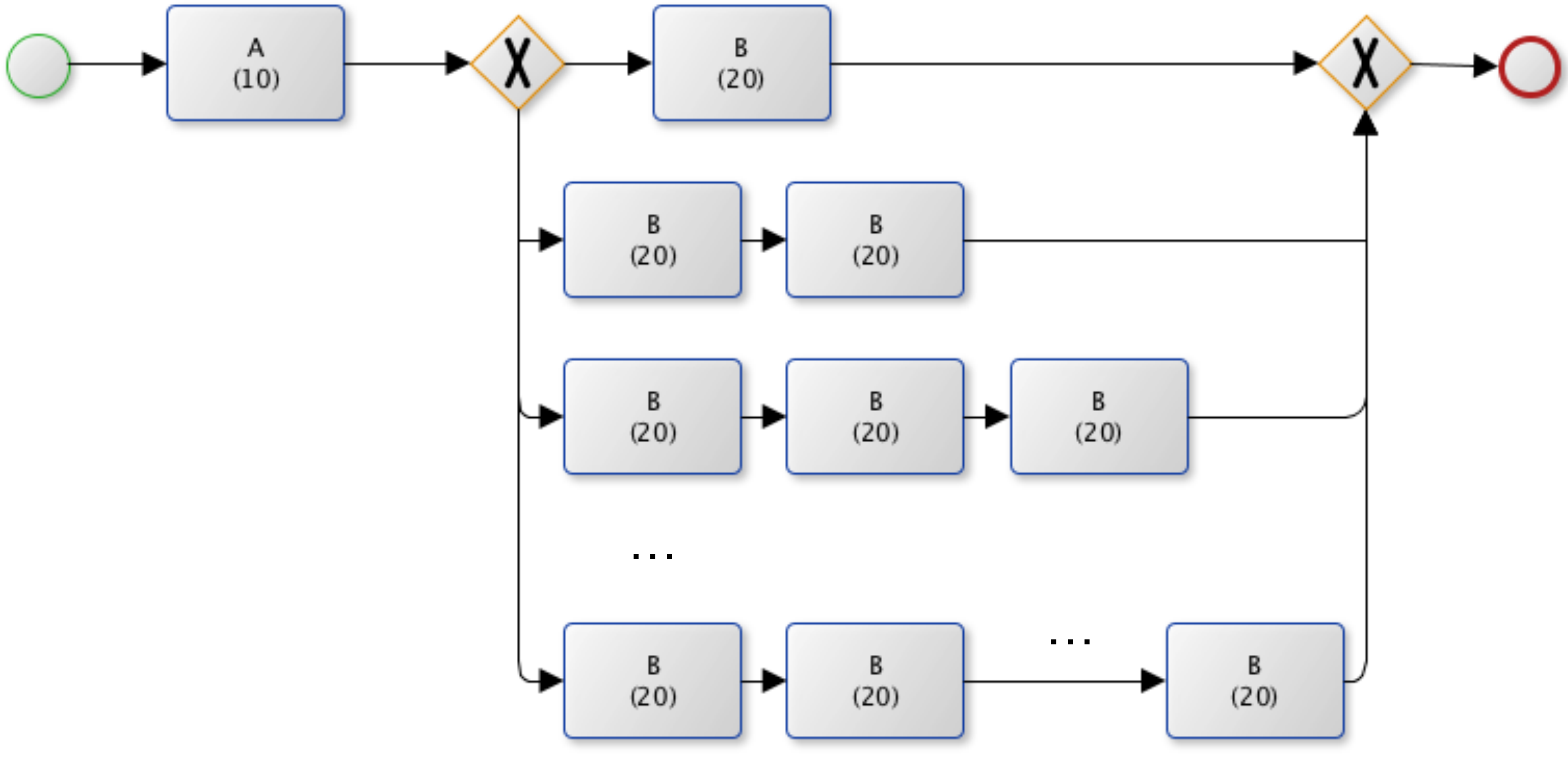
CT = 10 + 20 + 20 + 20 + … ?

Rework loop (1 or more times)



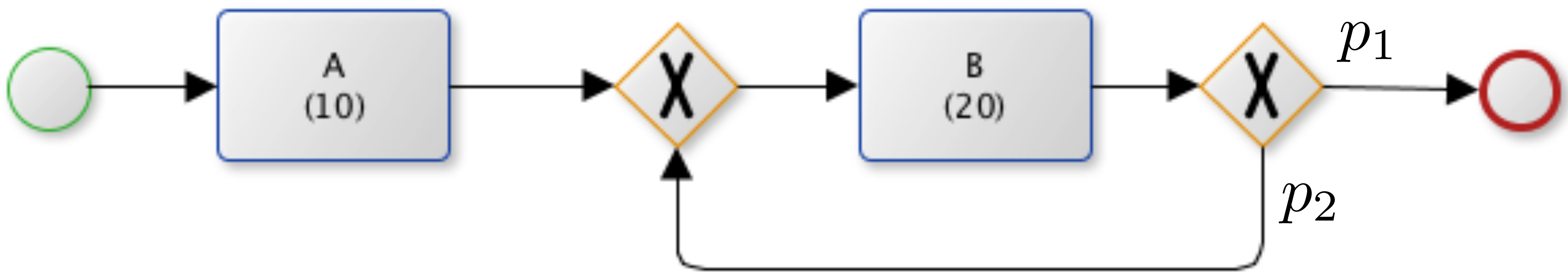
For sure we can say that B will be executed once.
($CT \geq 10 + 20 = 30$)
Then, depending on a choice, B can be executed twice.
Then, a third time, and so on …

Rework loop



For sure we can say that B will be executed once.
Then, depending on a choice, B can be executed twice.
Then, a third time, and so on …

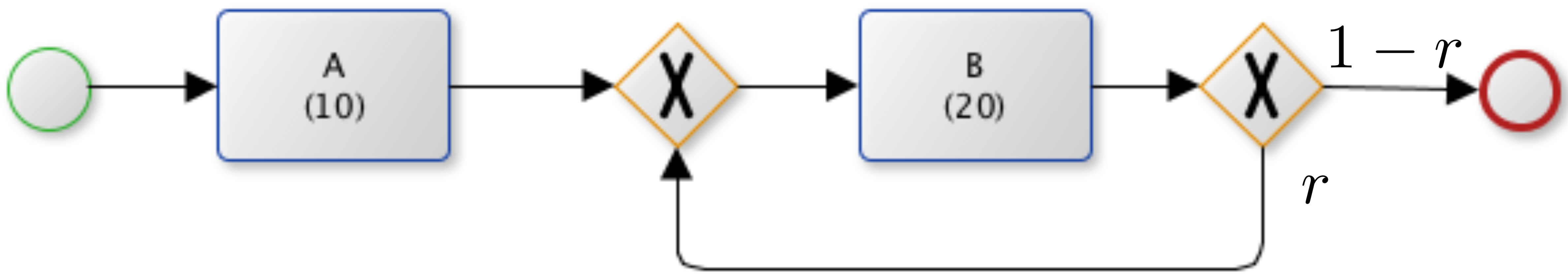
Branching probability, again...



$$p_1 + p_2 = 1$$

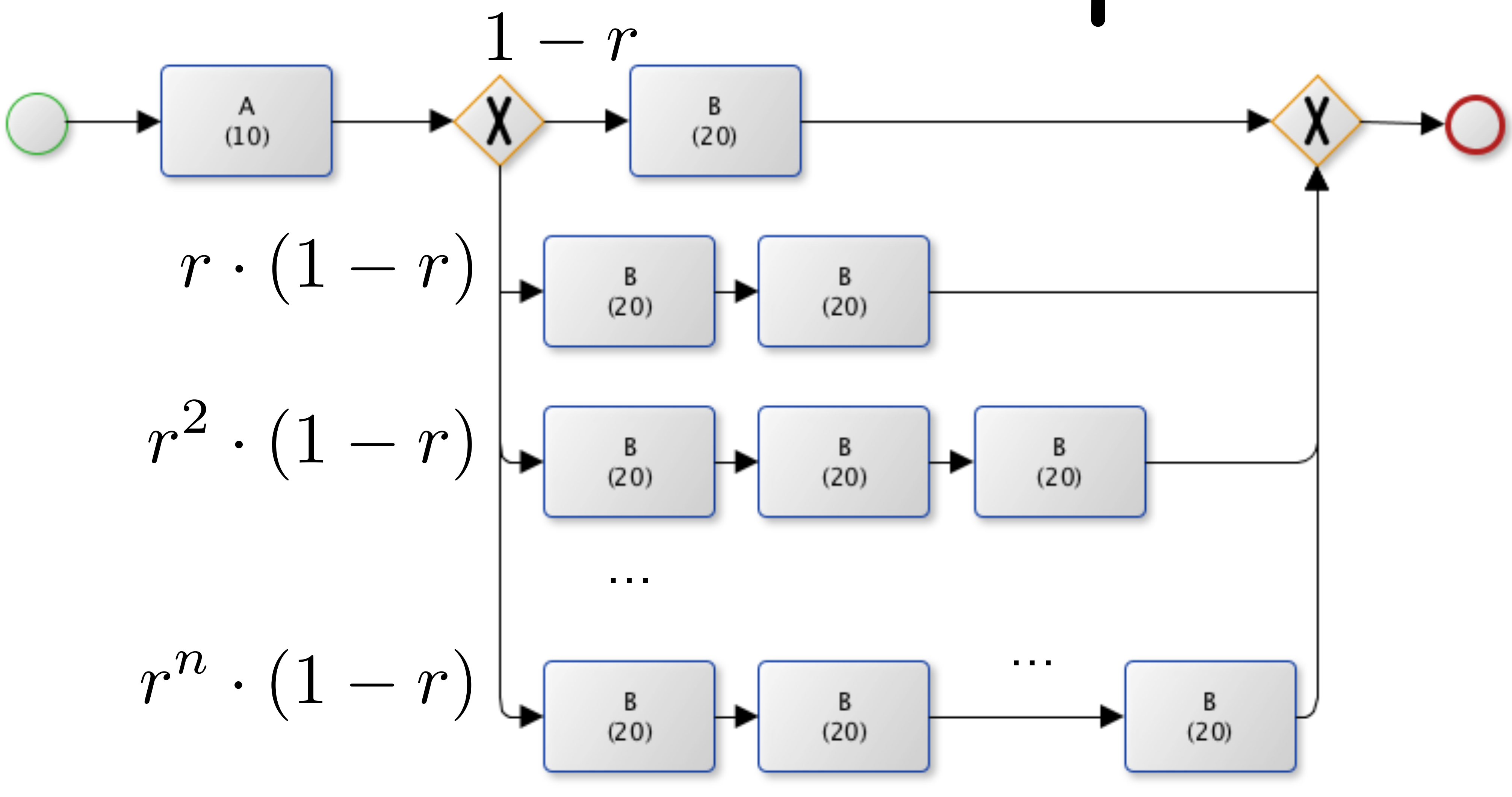
Branching probability p_i: is the frequency with which a given branch of a decision gateway is taken

Rework probability



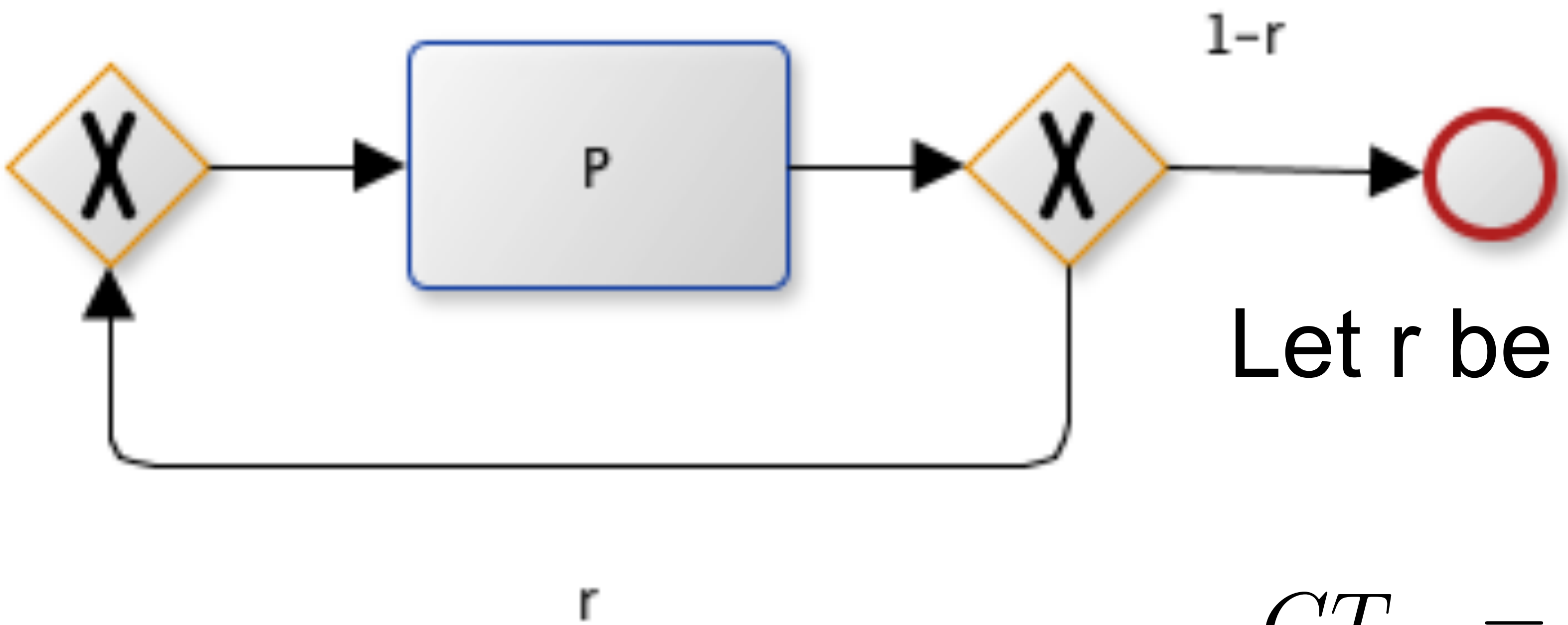
Rework probability r:
is the frequency with which the task is reworked

Rework loop



For sure we can say that B will be executed once.
Then, depending on a choice, B can be executed twice.
Then, a third time, and so on …
but always with less and less probability

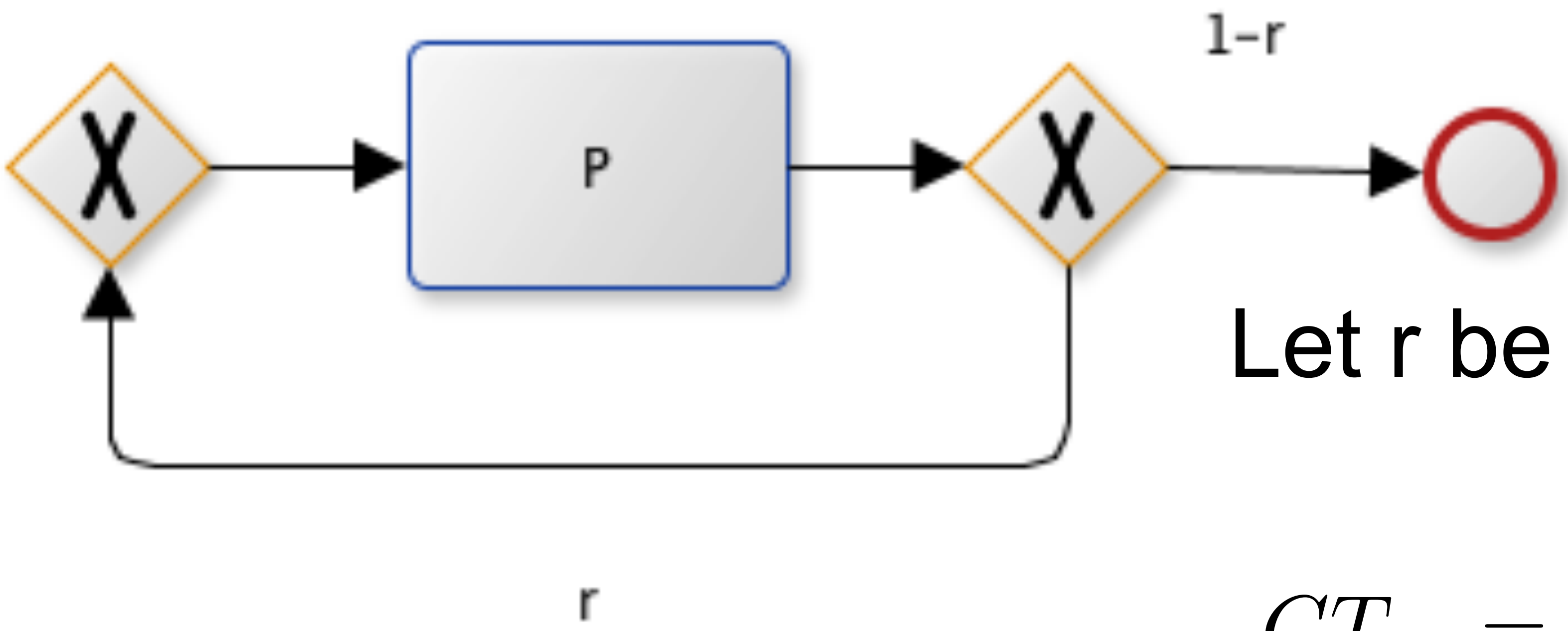
Rework loop



Let r be the *rework probability*, then:

$$
\begin{aligned}
CT &= 1 \cdot CT_P \cdot r^0 \cdot (1-r) \\
&+ 2 \cdot CT_P \cdot r^1 \cdot (1-r) \\
&+ \ldots \\
&+ n \cdot CT_P \cdot r^{n-1} \cdot (1-r) \\
&+ \ldots \\
&= \sum_{i=1}^{\infty} i \cdot CT_P \cdot r^{i-1} \cdot (1-r)
\end{aligned}
$$

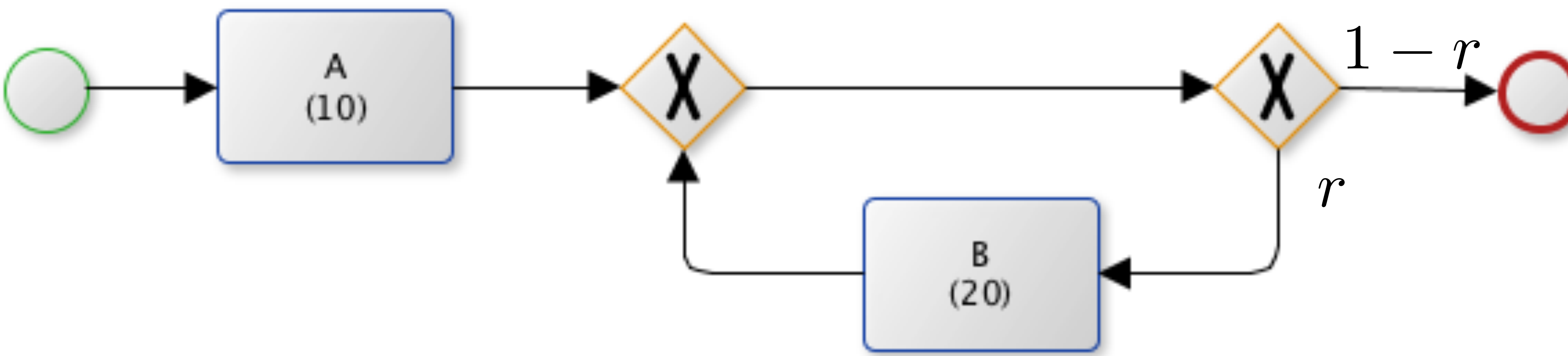
Rework loop



Let r be the *rework probability*, then:

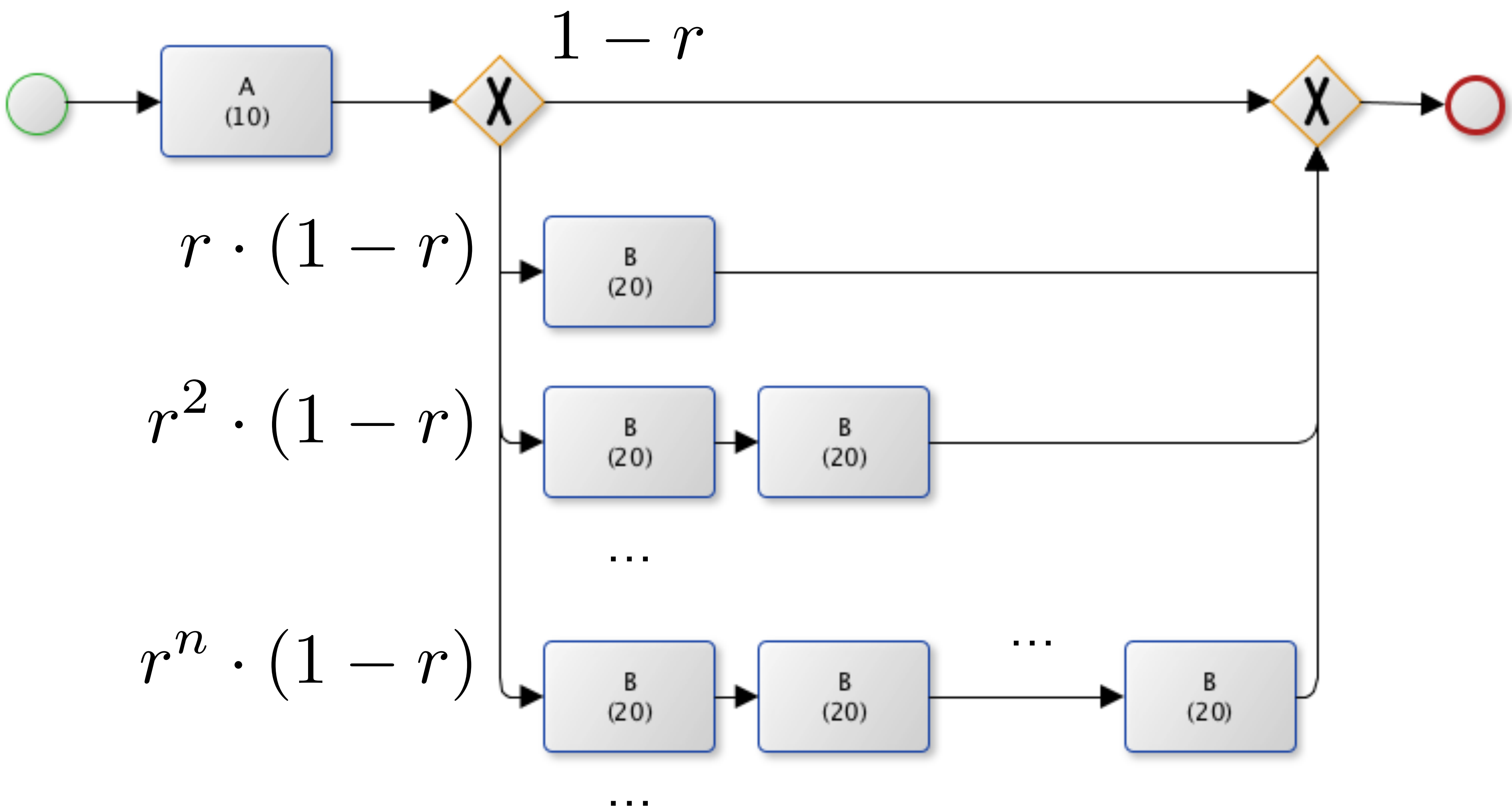
$$
\begin{aligned}
CT &= \sum_{i=1}^{\infty} i \cdot CT_P \cdot r^{i-1} \cdot (1-r) \\
&= CT_P \cdot (1-r) \cdot \sum_{i=1}^{\infty} i \cdot r^{i-1} \\
&= CT_P \cdot (1-r) \cdot \frac{1}{(1-r)^2} \\
&= \frac{CT_P}{1-r}
\end{aligned}
$$

Rework loop
(0 or more times)

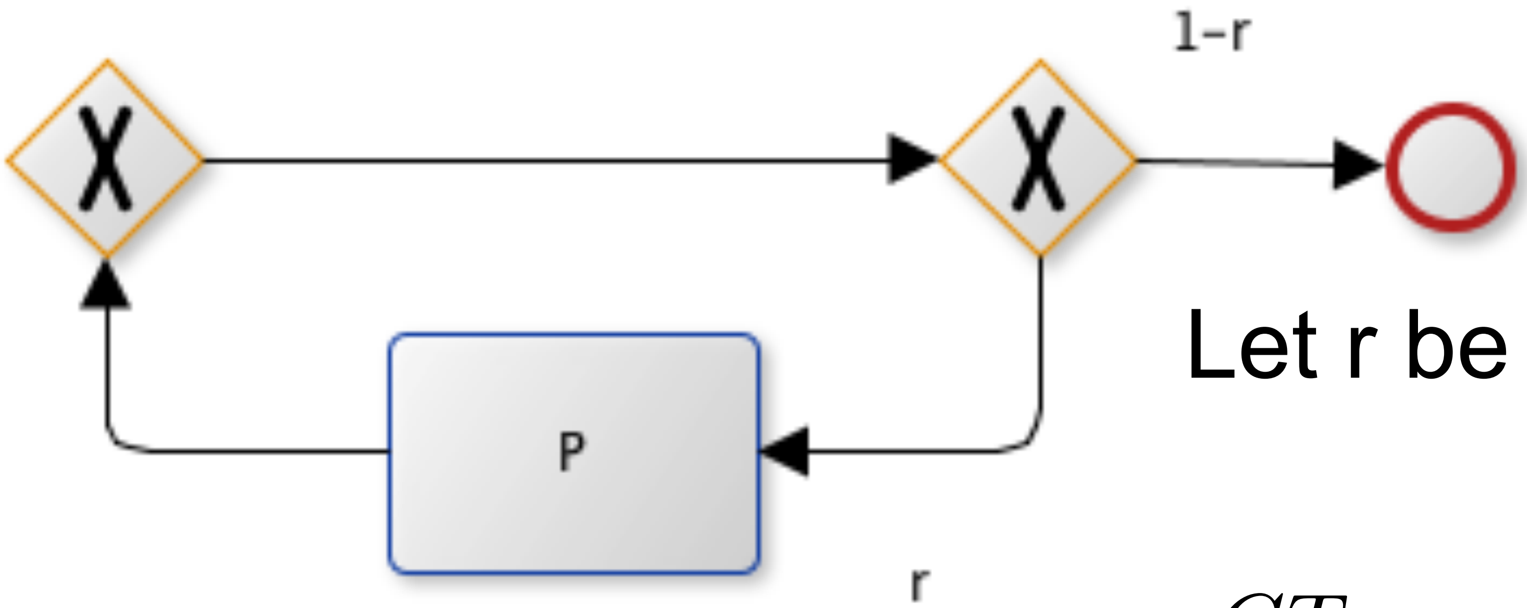


CT = ?

Rework loop



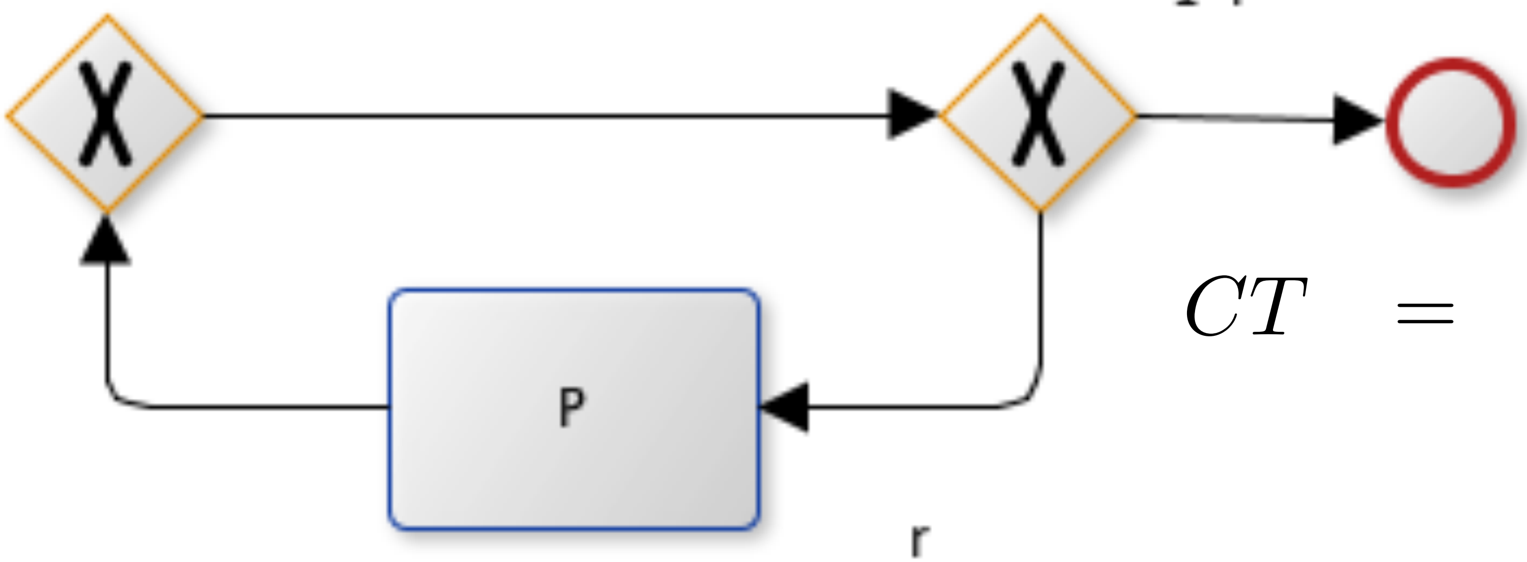
Rework loop



Let r be the *rework probability*, then:

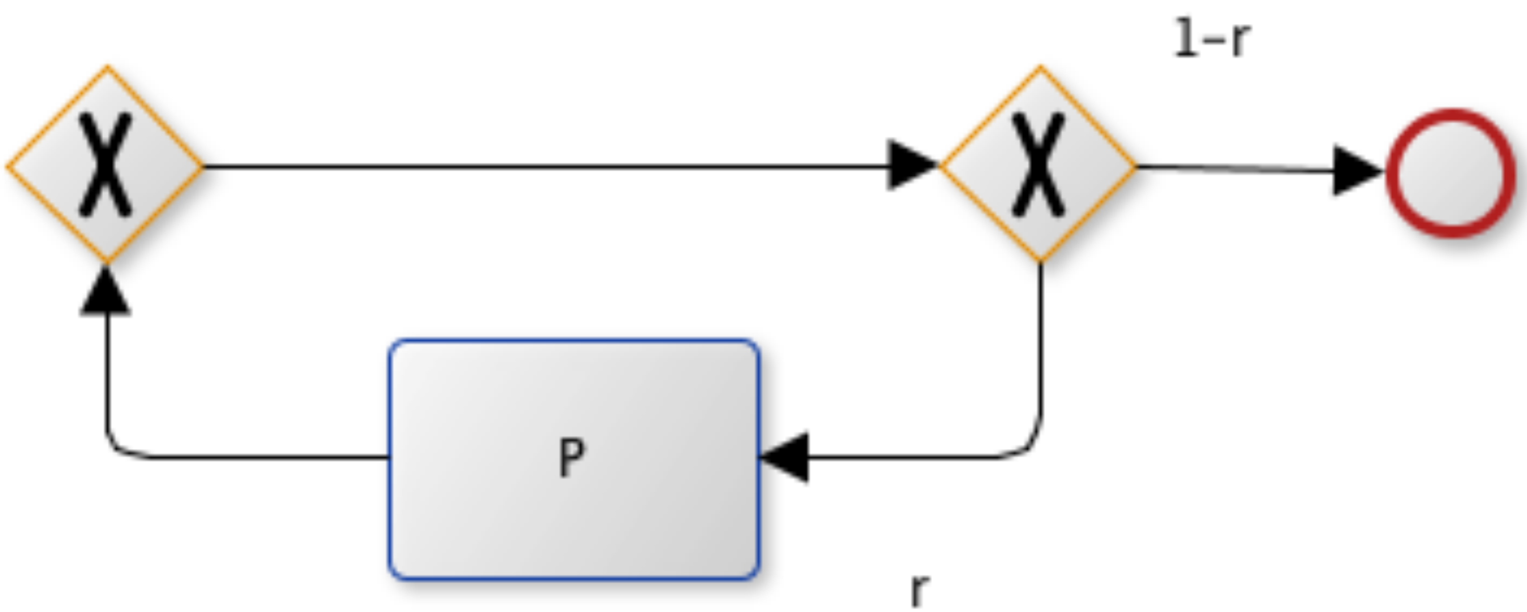
$$
\begin{aligned}
CT &= 0 \cdot CT_P \cdot r^0 \cdot (1-r) \\
&+ 1 \cdot CT_P \cdot r^1 \cdot (1-r) \\
&+ \ldots \\
&+ n \cdot CT_P \cdot r^{n-1} \cdot (1-r) \\
&+ \ldots \\
&= \sum_{i=0}^{\infty} i \cdot CT_P \cdot r^i \cdot (1-r)
\end{aligned}
$$

Rework loop



$$
\begin{aligned}
CT &= \sum_{i=0}^{\infty} i \cdot CT_P \cdot r^i \cdot (1-r) \\
&= \sum_{i=1}^{\infty} i \cdot CT_P \cdot r^i \cdot (1-r) \\
&= CT_P \cdot r \cdot (1-r) \cdot \sum_{i=1}^{\infty} i \cdot r^{i-1} \\
&= CT_P \cdot r \cdot (1-r) \cdot \frac{1}{(1-r)^2} \\
&= \frac{r \cdot CT_P}{1-r}
\end{aligned}
$$

Rework loop



Intuitively,

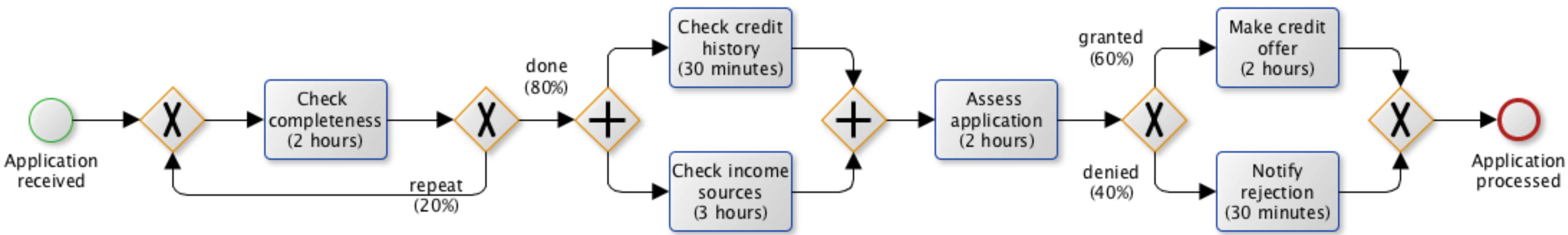
if $\dfrac{CT_P}{1-r}$ is the average cycle time for reworking P one or more times,

then $\dfrac{CT_P}{1-r} - CT_P = \dfrac{(1-(1-r)) \cdot CT_P}{1-r} = \dfrac{r \cdot CT_P}{1-r}$

is the average cycle time for reworking P zero or more times

Example

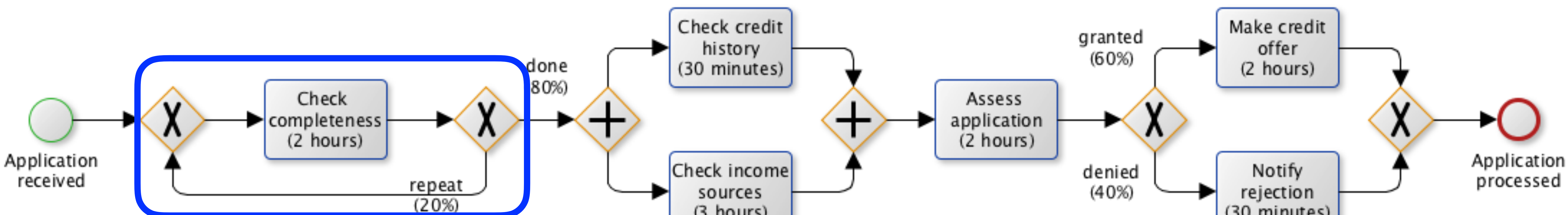
Compute the average cycle time CT of the process below



$$CT = \frac{2h}{1-0.2} + \max\{0.5h, 3h\} + 2h + 0.6 \cdot 2h + 0.4 \cdot 0.5h$$

Example

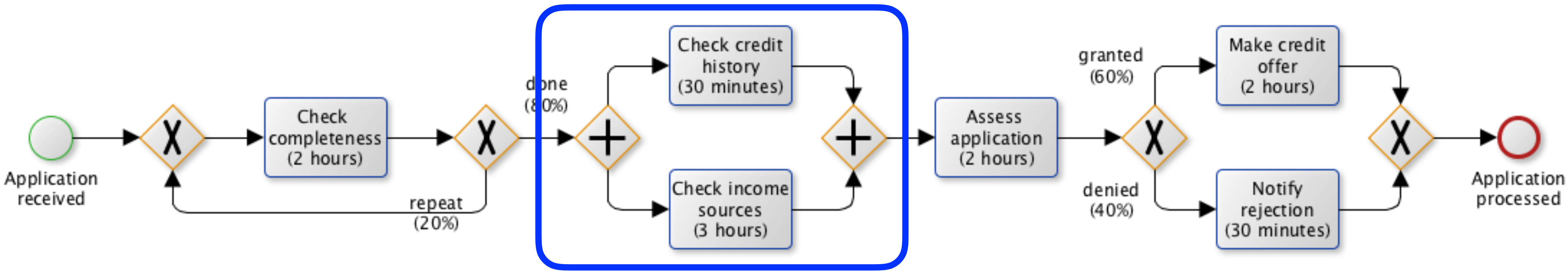
Compute the average cycle time CT of the process below



$$CT = \boxed{\frac{2h}{1-0.2}} + \max\{0.5h, 3h\} + 2h + 0.6 \cdot 2h + 0.4 \cdot 0.5h$$

Example

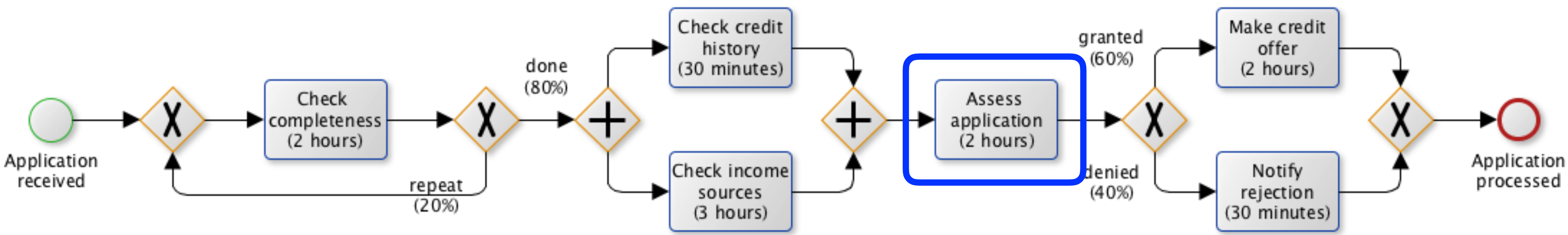
Compute the average cycle time CT of the process below



$$CT = \frac{2h}{1-0.2} + \boxed{\max\{0.5h, 3h\}} + 2h + 0.6 \cdot 2h + 0.4 \cdot 0.5h$$

Example

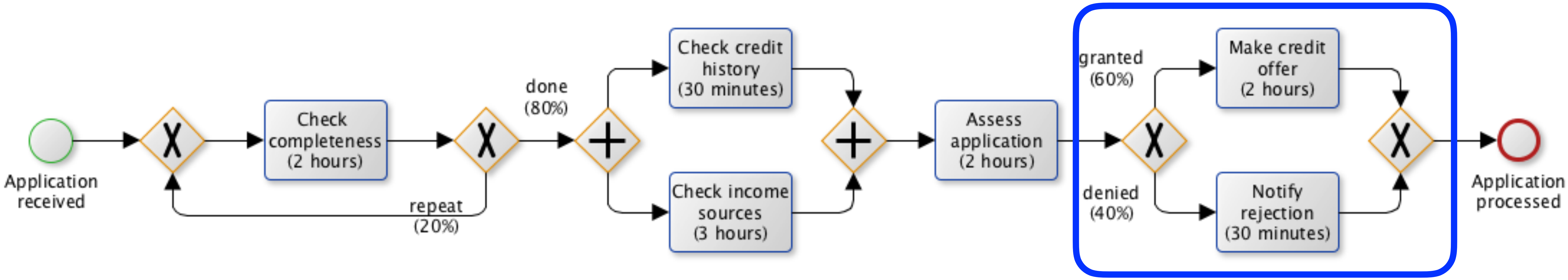
Compute the average cycle time CT of the process below



$$CT = \frac{2h}{1-0.2} + \max\{0.5h, 3h\} + \boxed{2h} + 0.6 \cdot 2h + 0.4 \cdot 0.5h$$

Example

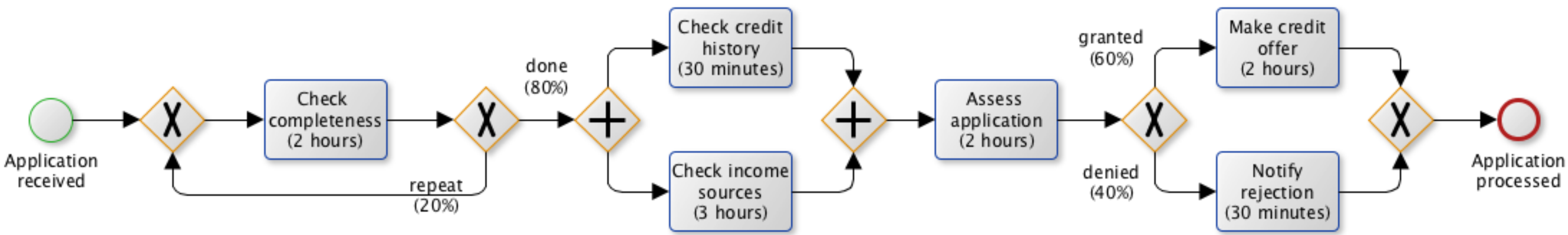
Compute the average cycle time CT of the process below



$$CT = \frac{2h}{1-0.2} + \max\{0.5h, 3h\} + 2h + \boxed{0.6 \cdot 2h + 0.4 \cdot 0.5h}$$

Example

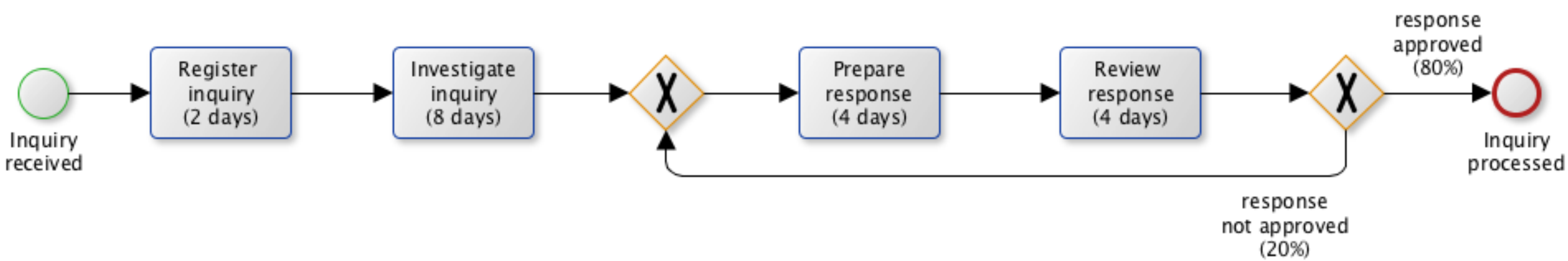
Compute the average cycle time CT of the process below



$$
\begin{aligned}
CT &= \frac{2h}{1-0.2} + \max\{0.5h, 3h\} + 2h + 0.6 \cdot 2h + 0.4 \cdot 0.5h \\
&= \frac{2h}{0.8} + 3h + 2h + 1.2h + 0.2h \\
&= 2.5h + 6.4h = 8.9h
\end{aligned}
$$

Exercise

Compute the average cycle time CT of the process below



Waiting vs processing

As mentioned at the beginning, the cycle time of an activity or a process can be divided into *waiting time* and *processing time*

Waiting time:
is the portion of the cycle time where no work is being done to advance the process (e.g. time spent in transferring documents or waiting for an actor to perform the work)

Processing time:
is the time that actors spend doing actual work

Waiting vs processing

In most processes,
the waiting time is a considerable portion of the cycle time!

For example,
in many situations cases are processed in batches
(e.g. applications, surveys)

and in many other cases actors are just not ready
(e.g. supervisor approval, medical prescription)

Theoretical cycle time

Assume that for each activity of the process both
the processing time and the cycle time are known

Let **TCT** denote the **theoretical cycle time** of the process:
this is computed in the same ways as CT,
but using the processing time of activities
(it is the amount of time a process would take on average
if no waiting time was necessary)

Cycle time efficiency

Cycle time efficiency (CTE):
is the ratio of processing time relative to the cycle time

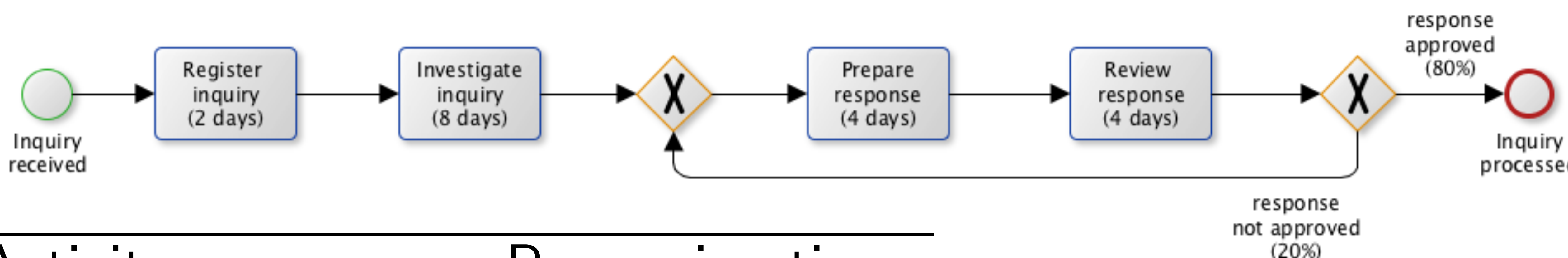
$$CTE = \frac{TCT}{CT}$$

A ratio close to 1 indicates that
there is little room for improving the cycle time
(unless radical changes in the process)

A ratio close to zero indicates that
there a significant amount of room for improving cycle time
(by reducing the waiting time)

Exercise

Compute the TCT and CTE of the process below, given the processing times reported in the table (assume 1 day = 8 working hours)



Activity	Processing time
Register inquiry	30 minutes
Investigate inquiry	12 hours
Prepare response	4 hours
Review response	2 hours

Limitation of flow analysis

Pitfalls and limitation

The equations we have presented
deals with *block-structured* process only:
we cannot calculate the cycle time for any processes.

The equations exploit the average cycle time of activities:
we need to estimate such values
(interviewing stakeholders, inspecting logs)

Flow analysis does not account for the fact that a process
can behave differently depending on the load:
when the load goes up and the resources are constant,
the waiting time increases.

Little's law

Arrival rate and Work-In-Process

Cycle time is directly related to two other important measures:

Arrival rate λ of a process:
is the average number of new instances of the process (i.e. cases) that are created per time unit

Work-In-Process (WIP):
is the average number of process instances (i.e. cases) that are active (i.e. not yet completed) at a given point in time

Little's law

In a paper from 1954,
operation research professor John Little
assumed (without giving a proof) that
the following equality holds in a *stable** system:

the long-term average number of customers (**WIP**)
is equal to
the long-term average effective arrival rate (**λ**) multiplied
by the average time a customer spends in the system (**CT**)

algebraically: $\mathbf{WIP = \lambda \cdot CT}$

* *stable* means that the number of customers in the system is not increasing infinitely

$$WIP = \lambda \cdot CT$$

Little's law tell us that:

WIP increases
if the cycle time (CT) increases
or if the arrival rate (λ) increases
(if the process slow down there will be more active cases and the faster new cases are created the higher will be the number of active instances)

If the arrival rate (λ) increases and we want to keep WIP constant, then we must decrease the cycle time (CT)
(i.e., we must work faster)

A note on Little's law

The law is classically stated using different symbols

L = λ · W

In a subsequent paper from 1961,
John Little proved the equality
later followed by simpler proofs in 1967, 1969, 1972

Since we can estimate WIP and λ by observing the system, we can use Little's law as an alternative way to calculate the average cycle time CT:

$$CT = \frac{WIP}{\lambda}$$

Example

Assume there are 250 business days per year.

If the total number of applications received over the last year is 2500 we can infer that the average number of *applications per day* is 10 (i.e. λ=10).

By sampling (e.g. checking every week), we observed that on average there were 200 *applications concurrently active* (i.e. WIP=200).

$$CT = \frac{WIP}{\lambda} = \frac{200}{10} = 20\,\mathrm{days}$$

Exercise

A restaurant receives on average 1200 customers per day (from 10am to 10pm).

During peak times (12pm to 3pm, and 6pm to 9pm) the restaurant receives around 900 customers and, on average, 90 customers can be found in the restaurant at a given time.

At non-peak times, the restaurant receives 300 customers in total and, on average, 30 customers can be found in the restaurant at a given time.

What are the average times that a customer spends in the restaurant during peak/non-peak times?

Exercise (continued)

The maximum capacity of the restaurant is sometimes reached during peak times.

The restaurant manager expects that the number of customers during peak times will increase slightly in the coming months.

What action can be taken to address this issue without investing in extending the building?

Cost analysis

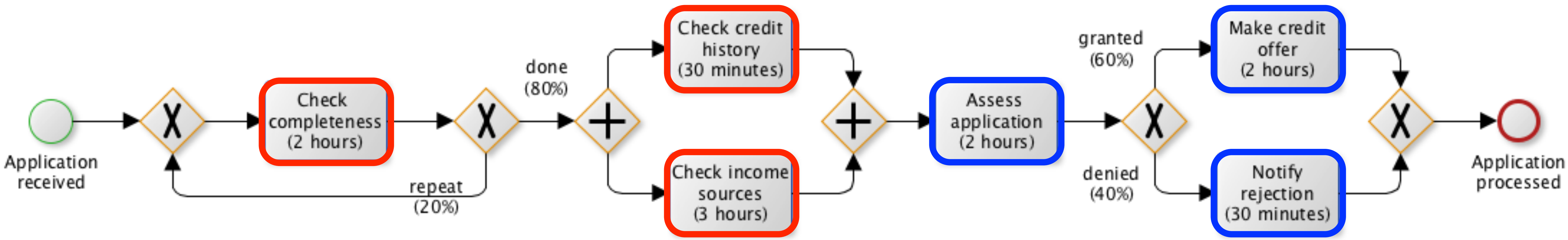
Cost analysis

Analogously to the case of cycle time computation,
flow analysis can be used to calculate other
performance measures.

If we know the average cost of each activity,
then we can calculate the average cost of the process
more or less as we have just seen.

In fact the formulas for
sequences, XOR-blocks and reworks are the same, but
for AND-blocks we need to take the sum (instead of max)

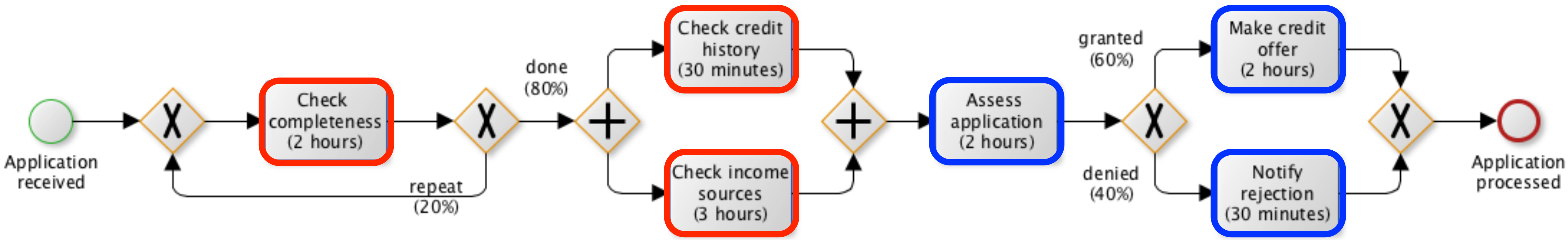
Example



Assume
activities are annotated with processing time,
"red" activities are performed by a clerk (hourly cost **25€**),
while "blue" activities by a credit officer (hourly cost **50€**).
Assume also that
the bank is charged **1€** for each ``credit history check''.

What is the average cost of the process?

Example (continued)

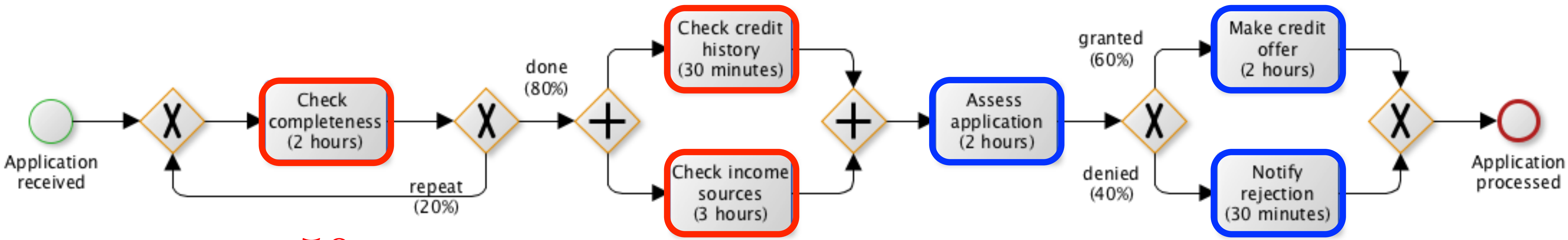


We can distinguish two kinds of costs:

human resource costs: can be calculated as the product of the (hourly) cost and the processing time of the task

other costs: fixed costs that are incurred by an execution of a task (not related to the time spent by human resources)

Example (continued)

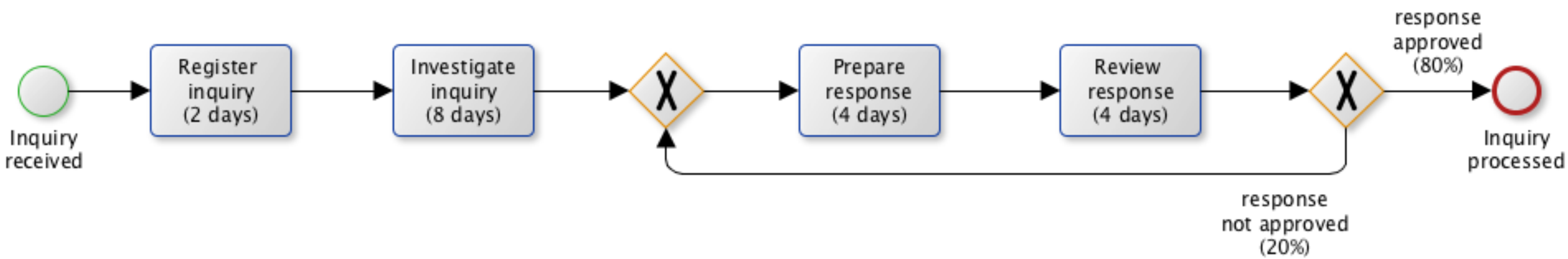


$$Cost = \frac{50}{1-0.2} + 13.5 + 75 + 100 + 0.6 \cdot 100 + 0.4 \cdot 25 = 321$$

Activity	Resource cost	Other cost	Total cost
Check completeness	$2 \cdot 25 = 50$		50
Check credit history	$0.5 \cdot 25 = 12.5$	1	13.5
Check income resources	$3 \cdot 25 = 75$		75
Assess application	$2 \cdot 50 = 100$		100
Make credit offer	$2 \cdot 50 = 100$		100
Notify rejection	$0.5 \cdot 50 = 25$		25

Exercise

Compute the average total cost of the process below



Activity	Resource	Resource hourly cost	Processing time
Register inquiry	clerk	25 euros	30 minutes
Investigate inquiry	advisor	50 euros	12 hours
Prepare response	senior advisor	75 euros	4 hours
Review response	counselor	100 euros	2 hours